Author: Pelt D.M.
Title: Filter-based reconstruction methods for tomography
Issue Date: 2016-05-03

Filter-based reconstruction methods for tomography

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C. J. J. M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 3 mei 2016
klokke 16:15 uur

door

Daniël Maria Pelt

geboren te Utrecht
in 1986

Promotor: Prof. dr. K. J. Batenburg

Samenstelling van de promotiecommissie:

Voorzitter: Prof. dr. A. W. van der Vaart
Secretaris: Prof. dr. S. J. Edixhoven

Overige leden: Prof. dr. S. Bals (Universiteit Antwerpen)
Prof. dr. R. H. Bisseling (Universiteit Utrecht)
Dr. F. de Carlo (Argonne National Laboratory)

Filter-based reconstruction methods for tomography

Daniël M. Pelt

The image on the cover shows the result of a single backprojection applied to eight projections of a circle with a width of 257 pixels and a uniform attenuation factor. This operation is similar to those involved in the computation of the filters introduced in Chapters 3 and 4 of this thesis. The image was computed by the ASTRA toolbox through its integration with TomoPy, introduced in Chapter 7.

The research in this thesis has been financially supported by the Netherlands Organisation for Scientific Research (NWO), programme 639.072.005. It was carried out at Centrum Wiskunde & Informatica (CWI), Amsterdam. Networking support was provided by COST Action MP1207.

Printed by: Gildeprint - Enschede
ISBN: 978-94-6233-269-0

Contents

1 Introduction and outline **1**
1.1 Tomography 1
1.2 Tomographic reconstruction 3
1.3 Overview of this thesis 8

2 Data-dependent filtering **11**
2.1 Introduction 11
2.2 Notation and concepts 13
2.3 Minimum residual filtered backprojection 17
2.4 Implementation 19
2.5 Additional constraints 21
2.6 Experiments 23
2.7 Results and discussion 24
2.8 Conclusion 33

3 Approximating SIRT by filtered backprojection **35**
3.1 Introduction 35
3.2 Method 36
3.3 Experiments 38
3.4 Conclusion 40

4 Local approximation of advanced regularized iterative methods **41**
4.1 Introduction 41
4.2 Notation and concepts 43
4.3 Method 48
4.4 Experiments 54
4.5 Results 55
4.6 Conclusions 63

5 Neural network filtered backprojection **65**
5.1 Introduction 65
5.2 Notation and concepts 67
5.3 Neural network filtered backprojection 72
5.4 Implementation 76
5.5 Experiments 79

5.6 Results and discussion 82
5.7 Conclusion 90

6 Application of NN-FBP in Electron Tomography 93
6.1 Introduction 93
6.2 Neural network filtered backprojection method 94
6.3 Results 98
6.4 Conclusion 105

7 Integrating TomoPy and the ASTRA toolbox 107
7.1 Introduction 107
7.2 Integrating TomoPy and the ASTRA toolbox 108
7.3 Installation and usage 111
7.4 Example 115
7.5 Conclusions 118

Bibliography 119

List of publications 131

Samenvatting 133

Curriculum Vitae 137

Acknowledgments 139

1

Introduction and outline

In this chapter, an introduction is given to tomography and the problem of tomographic reconstruction. A mathematical formulation of the problem is given, and several standard solution methods are explained. We will discuss the need for developing new efficient and accurate methods for use in modern tomographic experiments where the standard methods fail to produce sufficiently accurate results, and more advanced methods have computational costs that are too high to be used in practice. Finally, an overview is given of the main results of this thesis, in which several efficient and accurate methods are introduced.

1.1 Tomography

In many applications, it is useful to have a way of looking inside an object without destroying it. For example, in medical applications, being able to examine the internals of a patient without needing surgery is helpful for diagnosis. *Radiography* is often used in hospitals to perform this task. In radiography, a patient is briefly illuminated by a source of X-rays, and the rays passing through the patient are collected by a detector. Since different parts of the body absorb different amounts of X-rays that pass through, the resulting image on the detector shows the internal structure of the body. Bones, for example, are highly absorbing, and as such are usually clearly visible in the radiograph. An example of a radiograph is shown in Fig. 1.1.

The result of radiography is a two-dimensional (2D) image of the three-dimensional (3D) internal structure of the patient. Information about the structure in the direction of the X-rays is lost: in a radiograph, it is impossible to see whether a certain feature is located at the front or the back of the patient. Specialists are able to correctly diagnose patients using a 2D radiograph in many cases, since a lot of information is known about the human body. In other cases, however, the complete 3D information about the

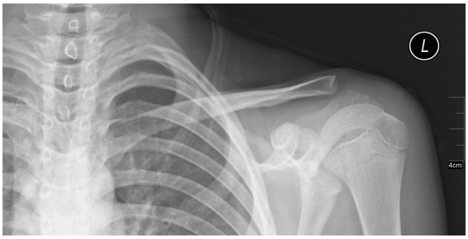


Figure 1.1: Radiograph of a human shoulder (© Nevit Dilmen).

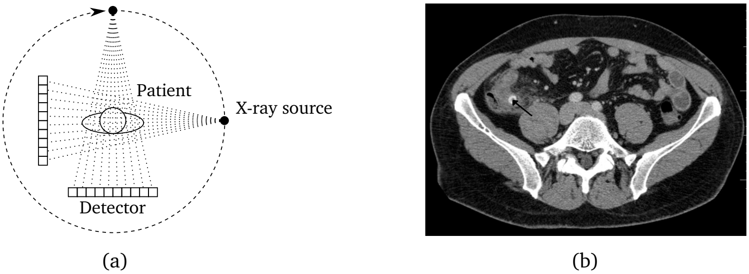


Figure 1.2: (a) Schematic overview of a computed tomography scanner. (b) A single slice of a reconstructed tomographic dataset (source: James Heilman, MD).

patient is needed. For example, when diagnosing or treating a tumor, it is important to know the exact position of the tumor in the body. *Computed tomography* is often used in these cases to acquire a 3D view of the internals of a patient.

In computed tomography, multiple radiographs are acquired while rotating the X-ray source and detector around the patient. In this way, X-ray images, called *projections* in this context, are acquired for several angles. A schematic overview of a computed tomography scanner is shown in Fig. 1.2a. The acquired 2D projections can be used to compute the 3D internal structure of the patient by a mathematical process called *tomographic reconstruction*. As an example, a single slice of a reconstructed 3D image of a patient's internals is shown in Fig. 1.2b. Various algorithms can be used to reconstruct tomographic data, which will be explained in more detail in Section 1.2. Tomography is used routinely in many applications other than medical diagnosis. Examples include material science [Sal+03], biomedical research [Lov+13], and industrial applications [Sip93]. A wide variety of radiation sources and scanning devices can be used, with length scales ranging from the nanoscale in electron tomography [Sco+12] to the astronomical scale in astrotomography [BSC01].

In this thesis, we will focus mainly on tomographic datasets acquired at a synchrotron radiation facility and datasets acquired with an electron microscope. Synchrotron radiation facilities, or *synchrotrons*, accelerate electrons in a circular path that can be several kilometers in length by applying strong magnetic fields at specific points along the path. At these points, a highly intense beam of photons is produced by the interaction of the high-energy electrons with the magnetic field. The photon beam has many useful properties for tomographic experiments, such as a high energy, flux, brilliance, and stability, and synchrotrons are routinely being used for advanced high-resolution tomographic experiments at the μm scale. Electron microscopes use a beam of electrons instead of photons to produce images. Since the wavelength of electrons is much smaller than the wavelength of X-rays, electron microscopes are able to image much smaller features compared to X-ray scanners. To perform *electron tomography*, i.e. tomographic experiments with electron microscopes, the sample is usually mounted on a holder that can rotate the sample within the beam. Because of physical restrictions of the system, the angular range for which projections can be acquired is typically limited in electron tomography. In most tomographic experiments performed with either synchrotrons or electron microscopes, the X-ray or electron beam does not diverge significantly when passing through the object, and can be regarded as a *parallel* beam. Mathematically, the fact that the beam is non-diverging has advantageous properties that we will exploit in this thesis.

1.2 Tomographic reconstruction

An important part of tomography is the reconstruction of the 3D structure using the acquired 2D projection images. In this section, we will define the reconstruction problem mathematically and give a brief overview of standard tomographic reconstruction methods. We will restrict the explanation to 2D parallel-beam problems, where the goal is to reconstruct a 2D image from one-dimensional (1D) projections, also called sinograms in this context. Note that 3D parallel-beam problems can usually be regarded as a collection of separate 2D parallel-beam problems for each slice.

Mathematically, we model the scanned object as a finite and integrable function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ with bounded support. We can define a single ray passing through the object as a line $l_{\theta,t}$ with the characteristic equation $t = x\cos\theta + y\sin\theta$. The line integral of f over a single line $l_{\theta,t}$, written as a function $P_\theta : \mathbb{R} \rightarrow \mathbb{R}$, is given by:

$$P_\theta(t) = \int_{l_{\theta,t}} f(x,y)\,\mathrm{d}s \tag{1.1}$$

$$= \iint_{\mathbb{R}^2} f(x,y)\delta(x\cos\theta + y\sin\theta - t)\,\mathrm{d}x\mathrm{d}y \tag{1.2}$$

Here, δ is the Dirac delta function. The goal of reconstruction in 2D parallel beam tomography is to recover the unknown function f given its line integrals $P_\theta(t)$ for different combinations of θ and t. The geometry is shown graphically in Fig. 1.3.

In practice, projection data are acquired only for a finite number of angles θ, using a finite set of detectors that each detect a single ray t per angle. If N_θ angles are used with N_d detectors per angle, the acquired dataset consists of $N_\theta N_d$ line integrals, one for each combination of angle $\theta \in \Theta = \{\theta_0, \ldots, \theta_{N_\theta - 1}\}$ and detector $d \in \{0, \ldots, N_d - 1\}$. The position τ_d of a detector d relative to the central detector is given by $\tau_d = s(d-(N_d-1)/2)$, and the set of detector positions by $T = \{\tau_0, \ldots, \tau_{N_d-1}\}$.

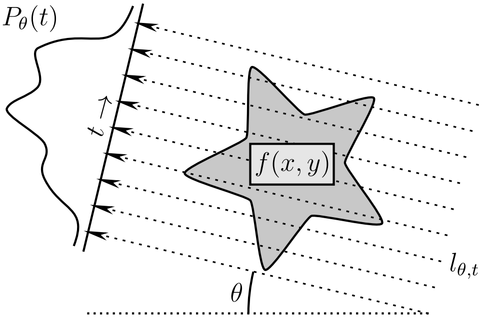


Figure 1.3: The two-dimensional parallel-beam tomography model. Parallel lines, rotated by angle θ, pass through the object f. A line $l_{\theta,t}$ has the characteristic equation $t = x\cos\theta + y\sin\theta$, and a projection $P_\theta(t)$ of f is given by the line integral of f over the line $l_{\theta,t}$.

In many cases, the unknown function f is reconstructed on a grid of $N \times N$ pixels, where N is often chosen to be equal to N_d.

We can write the acquired projections as a vector $\boldsymbol{p} \in \mathbb{R}^{N_\theta N_d}$ with $N_\theta N_d$ elements. Similarly, the reconstructed image can be written as a vector $\boldsymbol{x} \in \mathbb{R}^{N^2}$ with N^2 elements. Using these definitions, we can write the tomographic acquisition process as a linear equation:

$$\boldsymbol{p} = \boldsymbol{W}\boldsymbol{x} \tag{1.3}$$

Here, $\boldsymbol{W}$ is a $N_\theta N_d \times N^2$ matrix that corresponds to computing the line integrals of the object $\boldsymbol{x}$. In the matrix $\boldsymbol{W}$, the element w_{ij} specifies the contribution of pixel j to detector i. The multiplication of an image $\boldsymbol{x}$ by $\boldsymbol{W}$ is called the *forward projection* of $\boldsymbol{x}$, and the multiplication of a vector $\boldsymbol{p}$ by $\boldsymbol{W}^T$ is called the *backprojection* of $\boldsymbol{p}$. Note that for typical sizes of tomographic datasets, the dimensions of $\boldsymbol{W}$ will be too large to store it in computer memory. Instead, forward projections are usually computed on-the-fly by computing the line integrals of an image $\boldsymbol{x}$ directly [PBS11], with backprojections computed on-the-fly as well. The projection operations can be computed efficiently using graphic processor units (GPUs), which helps to reduce reconstruction times in practice [XM05; MXN07].

Two types of methods are commonly used to reconstruct the unknown object from the acquired projections: *analytical* reconstruction methods and *algebraic* reconstruction methods.

Analytical reconstruction

Analytical reconstruction methods are based on taking the continuous model of tomographic acquisition (Eq. (1.2)) and inverting it to find an expression for $f(x,y)$. The result of this approach for 2D parallel-beam tomography is the *filtered backprojection* method (FBP). FBP starts by convolving the acquired projection data $P_\theta(t)$ with a filter $h : \mathbb{R} \rightarrow \mathbb{R}$:

$$q_\theta(t) = \int_{-\infty}^{\infty} h(\tau) P_\theta(t-\tau) \mathrm{d}\tau \tag{1.4}$$

This convolution operation can be efficiently performed in Fourier space, with $\hat{h}$ and $\hat{P}_\theta$ being the Fourier transforms of h and P_θ:

$$q_\theta(t) = \int_{-\infty}^{\infty} \hat{h}(u)\hat{P}_\theta(u)e^{2\pi\imath ut}\mathrm{d}u \tag{1.5}$$

By taking $\hat{h}(u) = |u|$, we obtain an expression for $f(x, y)$ [KS01]:

$$f(x,y) = \int_0^{\pi} q_\theta(x\cos\theta + y\sin\theta)\mathrm{d}\theta \tag{1.6}$$

In practice, projections are only acquired for a finite number of angles θ and detectors τ, as explained above. Therefore, we have to discretize Eq. (1.6) to be able to obtain the filtered backprojection method:

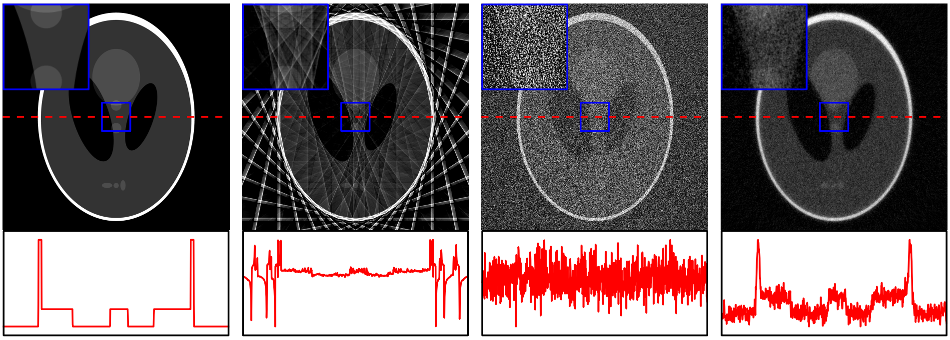
$$f(x,y) \approx FBP_h(x,y) = \sum_{\theta_d\in\Theta}\sum_{\tau_p\in T} h(\tau_p)P_{\theta_d}(t-\tau_p) \tag{1.7}$$

where $t = x\cos\theta + y\sin\theta$. Since $t-\tau_p$ is not necessarily equal to one of the acquired detector positions, some interpolation is needed to find the value of $P_{\theta_d}(t-\tau_p)$. Linear interpolation is often used, since projection data are usually reasonably smooth. The filter h is only needed at a finite number of discrete positions $h(\tau_p)$, and is often defined as a vector $\boldsymbol{h} \in \mathbb{R}^{N_d}$. Several standard filters are commonly used in practice, such as the Ram-Lak (ramp), Shepp-Logan, and Hann filters [Far+97; WWH05]. One of the most popular filters is the Ram-Lak filter, which is obtained by discretizing the theoretical $\hat{h}(u) = |u|$ filter. In the matrix and vector notation, the FBP method can be written as:

$$\boldsymbol{x}_{FBP} = FBP_{\boldsymbol{h}}(\boldsymbol{p}) = \boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{h}}\boldsymbol{p} \tag{1.8}$$

Here, $\boldsymbol{C}_{\boldsymbol{h}}$ is a convolution operation that convolves each 1D array of detector values, taken at a single rotation angle, with the 1D filter $\boldsymbol{h}$.

The filtered backprojection method is computationally efficient, since the convolution operation can be computed using the Fast Fourier Transform method, and only a single backprojection is needed to reconstruct an image. The reconstruction quality, however, depends on how well the discretized equation (Eq. (1.7)) approximates the continuous inverse equation (Eq. (1.6)). The continuous inverse equation assumes that noise-free projection data are available for an infinite number of rotation angles between 0 and π, which is not possible to achieve in practice. If only a limited number of projections are acquired or when there is noise present in the acquired projections, reconstructions computed with the FBP method tend to contain significant artifacts that can make further analysis impossible. Examples of typical artifacts found in FBP reconstructions are shown in Fig. 1.4. When sufficiently many projections are available and the noise in them is sufficiently small, however, the FBP method is usually able to produce reconstructions that are accurate enough for analysis. For this reason, and because of its computational efficiency, filtered backprojection remains the most popular reconstruction method in many applications of tomography [PSV09].



(a) (b) (c) (d)

Figure 1.4: Three reconstructions of the Shepp-Logan head phantom (a), showing artifacts that occur with imperfect data. In (b) and (c), FBP was used to reconstruct the images. In (b), data from only 16 projections were used, and severe streak artifacts are present in the result. In (c), data from 1024 projections were used, but a large amount of Poisson noise was present, resulting in severe noise in the resulting image. In (d), SIRT, an algebraic method, was used to reconstruct the image using the same noisy projection data as in (c). The algebraic reconstruction image (d) has less noise compared to the FBP reconstruction (c). Under each image the line profile of the middle row is shown, and a small section is shown enlarged in the upper left of each image.

Algebraic reconstruction

Algebraic reconstruction methods are based on the discrete matrix representation of the tomographic reconstruction problem (Eq. (1.3)). Specifically, the reconstruction problem is written as a system of linear equations, which is then solved by an optimization method. Most algebraic methods minimize the difference between the forward projection of the reconstructed image with the acquired projection data, which is called the *projection error*. A popular choice is to minimize the ℓ_2-norm of the projection error, in which case we can write the algebraic approach as the following optimization problem:

$$\boldsymbol{x}_{alg} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^{N^2}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2 \tag{1.9}$$

Because of the size of the matrix $\boldsymbol{W}$, it is often impossible to use direct methods such as singular value decomposition to find a solution to Eq. (1.9). Instead, methods are used in which the projection error is iteratively minimized. Different mathematical optimization methods can be used to minimize the projection error, leading to different algebraic reconstruction methods. For example, if the projection error is minimized by Landweber iteration [Lan51], i.e. using gradient-descent steps on the projection error, the result is the Simultaneous Iterative Reconstruction Technique (SIRT). Iterations of the SIRT method can be written as:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha \boldsymbol{W}^T\left(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}^k\right) \tag{1.10}$$

Here, $\alpha \in \mathbb{R}$ is a parameter that influences the convergence rate and stability of the method. Other examples of standard algebraic methods are the CGLS method, where the projection error is minimized by a Conjugate Gradient method [HS52], and the Algebraic Reconstruction Technique (ART) [GBH70], where the error is minimized by the Kaczmarz method [Kac37].

Since algebraic methods use a model of the tomographic reconstruction problem that includes only the projections that are actually acquired instead of assuming that an infinite number of projections are available, they tend to handle problems with a limited number of projections better than analytical methods. Furthermore, the effect of noise in the projection data can be limited in most algebraic methods by stopping the iteration process early, which is a form of implicit regularization. A comparison between a reconstruction computed by the algebraic SIRT method with a reconstruction computed by the analytical FBP method is shown in Fig. 1.4. Note that the artifacts caused by the noise are significantly reduced in the algebraic reconstruction, shown in Fig. 1.4d, compared to the analytical reconstruction, shown in Fig. 1.4c.

One of the main disadvantages of algebraic methods is their high computational costs. Several tens or hundreds of iterations are typically needed in algebraic methods to converge to an acceptable reconstruction image, with multiple projection operations per iteration. For the large datasets that are routinely acquired at experimental facilities, the high computational costs can be prohibitive in practice. For example, suppose we want to reconstruct a 1024^3 3D volume using 1024 projections of 1024×1024 pixels each, which would be a medium-sized problem in synchrotron tomography. Using a state-of-the-art GPU system, it would take around 80 seconds to reconstruct the full volume with the analytical FBP method, while reconstructing with 200 iterations of the algebraic SIRT method would take around one and a half hours. Since a single tomographic dataset can often be acquired in a few minutes or less, the reconstruction time of algebraic methods tends too be too long to routinely use them in practice, and FBP is used instead.

In advanced tomographic experiments, one is often restricted to acquiring only a very limited number of projections, and a large amount of noise can be present in each projection. In these cases, algebraic methods are often also unable to produce reconstructions that are sufficiently accurate for further analysis, similar to analytical methods. The reason for this is that the linear system that is solved in algebraic methods (Eq. (1.9)) is highly underdetermined if $N_\theta N_d \ll N^2$. The result is that there exist infinitely many reconstructions that have the same projection error, most of which are not suitable for analysis. It depends on the specific optimization method that is used which reconstruction is computed by an algebraic method. Furthermore, the projection matrix $\boldsymbol{W}$ is ill-conditioned in most applications of tomography. This means that even minor inconsistencies in the projection data, such as noise, can have a large effect on the reconstructed image.

The reconstruction quality of algebraic methods can be improved by exploiting prior knowledge about the scanned object or scanning system. Mathematically, this prior knowledge is often encoded as an additional term $g : \mathbb{R}^{N^2} \rightarrow \mathbb{R}$ in the objective function that is minimized, resulting in *regularized iterative methods*:

$$\boldsymbol{x}_{reg} = \underset{\boldsymbol{x} \in \mathbb{R}^{N^2}}{\operatorname{argmin}} \left[\| \boldsymbol{p} - \boldsymbol{W}\boldsymbol{x} \|_2 + \lambda g(\boldsymbol{x}) \right] \tag{1.11}$$

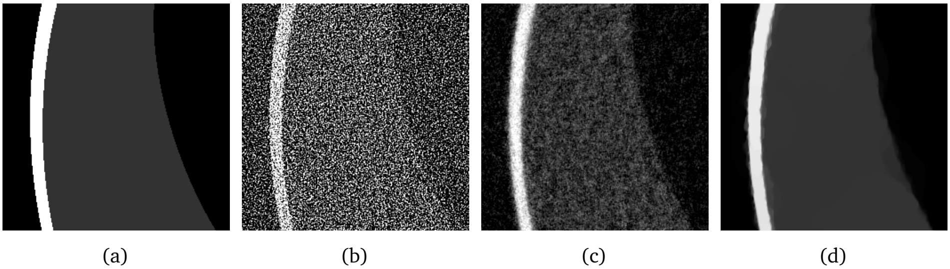

(a) (b) (c) (d)

Figure 1.5: Zoomed-in reconstructions of the Shepp-Logan head phantom (a), showing the resulting images of three different reconstruction methods: (b) FBP, (c) SIRT, and (d) total variation minimization. The images were reconstructed on a 1024 × 1024 pixel grid, using projection data acquired with $N_d = 1024$ detectors and $N_\theta = 256$ projection angles, equally distributed in the interval $[0, \pi]$, and additional Poisson noise applied.

Here, $g(\boldsymbol{x})$ is a penalty function that penalizes undesired solutions that do not fit with the assumed prior knowledge, and the λ term controls how strongly the penalty function is weighted compared to the projection error term. For example, when it is assumed that the gradient of the reconstructed object is sparse, a popular choice is to use *Total Variation minimization* (TV-minimization) by setting $g(\boldsymbol{x}) = \|\nabla \boldsymbol{x}\|_1$, where ∇ is a discrete gradient operator [SP08]. If the assumed prior knowledge prior knowledge is appropriate for the acquired data, regularized iterative methods can be extremely successful in reconstructing objects from (highly) limited data [BS11; Kos+13]. A comparison between FBP, SIRT, and TV-minimization reconstructions is shown in Fig. 1.5 for noisy projection data. A major disadvantage of regularized iterative methods is their high computational costs, which tend to be even higher than those of algebraic methods. For example, when reconstructing the 1024^3 volume defined above on the same state-of-the-art GPU system, it would take more than a *day* to reconstruct the full volume using the FISTA method for TV-minimization [BT09a].

1.3 Overview of this thesis

As explained above, in many applications of tomography, analytical methods produce reconstructions that are not accurate enough for further analysis. More accurate reconstructions can be obtained by using (regularized) iterative methods, but these can have computational costs that are too high to be used in practice. In this thesis, new reconstruction methods are developed that combine the analytical and algebraic approaches, resulting in methods that are as computationally efficient as analytical methods, but with a reconstruction accuracy of algebraic methods. Analytical methods allow for changing the filter $\boldsymbol{h}$ without increasing the needed computation time. We will use this freedom in filter choice to develop new *filter-based* reconstruction methods, which are based on the FBP method with specific filters. The filters can be defined and computed in different ways, and can depend on the acquisition geometry, the scanned object, and/or a separate pre-computing step. Several filter-based methods are introduced in this thesis, and reconstruction results are compared with other popular methods. In the rest of this section, the contributions and results of each chapter are explained.

In Chapter 2, the MR-FBP method is introduced, which uses a filter that minimizes the projection error of the resulting FBP reconstruction. The filter can be computed using an approach that is similar to algebraic reconstruction methods, i.e. minimizing the ℓ_2-norm of the projection error:

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}\in\mathbb{R}^{N_d}}{\operatorname{argmin}} \|\boldsymbol{p} - \boldsymbol{W}FBP_{\boldsymbol{h}}(\boldsymbol{p})\|_2 \tag{1.12}$$

Note that the filter that is computed depends on the acquired data. The results of Chapter 2 show that the method is able to produce reconstructions with similar reconstruction quality as the algebraic SIRT method, but is much faster at producing them. Furthermore, it is shown that some forms of prior knowledge can be exploited to improve reconstruction quality, and that the computed filters automatically adapt to the characteristics of the object and the scanning parameters.

A different approach is taken in Chapter 3, where the SIRT-FBP method is introduced. This method explicitly approximates the SIRT method by the FBP method with a specific filter. The approximation is achieved by first rewriting the SIRT method into a matrix form, and then approximating k iterations of SIRT by a single convolution operation. By comparing with the standard FBP method, it is possible to show that the approximated SIRT method is identical to the FBP method with a specific angle-dependent filter. The filter can be pre-computed for a certain scanning geometry by a single SIRT-like iteration method, after which it can be reused for datasets that are acquired with the same geometry. The results of Chapter 3 show that reconstructions computed with the SIRT-FBP method are virtually identical to standard SIRT reconstructions.

In Chapter 4, a method is introduced that allows one to approximate a slow regularized iterative method inside a (small) subvolume of the complete reconstruction volume. If one is only interested in a small subvolume of the scanned object, as is often the case, this method can significantly reduce the computation time that is needed to perform the required analyses. Note that regularized iterative methods generally need to compute the entire reconstruction volume, since they are based on minimizing a global objective function (Eq. (1.11)). The local approximation method is based on extending the SIRT-FBP method introduced in Chapter 3 to allow for different types of additional regularization terms. In the results of Chapter 4, we show that the reconstruction quality of the local approximations is almost identical to that of the much slower global regularized iterative methods for several popular types of prior knowledge, such as TV-minimization.

The NN-FBP method is introduced in Chapter 5. An NN-FBP reconstruction consists of a nonlinear combination of multiple FBP reconstructions, each with a different filter. The filters are pre-computed by using projection data and high-quality reconstructions of objects that are similar to the objects that will be reconstructed later. Methods from neural network theory are used to *train* the filters, after which they can be used to accurately reconstruct similar objects even when only a limited number of projections are acquired. The results of Chapter 5 show that the NN-FBP method is able to produce reconstructions with a significantly higher quality than both analytical and algebraic reconstruction methods. Also, it is shown that the method can be used to approximate slow regularized iterative methods in the case that it is not possible to acquire a large number of projections for use in training.

An application of the NN-FBP method in electron tomography is presented in Chapter 6. In electron tomography, acquiring a large number of projections is both time-consuming and labor-intensive. As a result, it is difficult to scan a large number of samples and obtain statistically significant results for certain sample characteristics. By lowering the number of projections that need to be acquired, the effort needed to obtain statistically significant results can be reduced. In Chapter 6, we use the NN-FBP method to obtain accurate reconstructions of gold nanoparticles using a very limited number of projections. The NN-FBP method is trained on a small number of nanoparticles that are scanned with a higher number of projections. Results show that the NN-FBP method is able to produce accurate reconstructions with only a few projections, and that the method can be used to obtain statistically significant results with reduced scanning time.

In Chapter 7, a different problem is addressed compared to the other chapters of this thesis. An additional reason that (regularized) iterative methods are not used routinely at experimental facilities is a practical one: it is often difficult to install and use software that can perform these reconstructions at the facilities. In Chapter 7, we improve this situation for the synchrotron community by combining a software toolbox that is specifically designed to be easy-to-use at synchrotrons, TomoPy [Gür+14], with a toolbox that can be used to develop advanced reconstruction methods, the ASTRA toolbox [Aar+15]. The result of this combination is that methods that are developed using the ASTRA toolbox can be easily installed and used at synchrotrons, with minimal changes needed in user scripts. In Chapter 7, some code examples are given, explaining the various capabilities and options of the integrated toolboxes. Furthermore, results are shown for the reduction of computation time that can be achieved for iterative methods when using the optimized GPU-based methods of the ASTRA toolbox compared to using CPU-based methods. Finally, an example for an experimental dataset is given, where the combination of the preprocessing methods of TomoPy and the reconstruction methods of the ASTRA toolbox is able to improve reconstruction quality in practice.

2

Data-dependent filtering

2.1 Introduction

Tomographic reconstruction problems are found in many applications, such as X-ray scanners in medical imaging, or electron microscopy in materials science [Gra13]. In the standard tomographic problem, we aim to reconstruct an object from its projections, acquired for a range of angles. This problem has been studied extensively because of its practical relevance, leading to a wide range of reconstruction methods. For an overview of previous work, see [KS01; Nat01; Buz08]. Most of the current reconstruction methods can be separated into two groups: *analytical* methods and *algebraic* methods.

The basis of analytical reconstruction methods is a continuous representation of the tomographic problem. This continuous model is inverted, and the result is discretized. The resulting reconstruction methods, of which the filtered backprojection (FBP) method is the most widely used, are usually computationally efficient. Furthermore, if projection data of sufficiently high quality is available, reconstructions computed by these methods are often accurate. These two properties are among the reasons that the FBP method is very popular in practice [PSV09], along with its ease of implementation. An important drawback of analytical methods is that they are based on an approximation of a model where perfect data is available for *all* projection angles. If the available data is not perfect, either because few projections are available or because the data is noisy, the quality of analytical reconstructions will suffer from interpolation effects.

Practical considerations can lead to limited or noisy projection data in many applications of tomography. In electron tomography, for example, the electron beam

---

This chapter is based on:

D. M. Pelt and K. J. Batenburg. "Improving Filtered Backprojection Reconstruction by Data-Dependent Filtering". *IEEE Transactions on Image Processing* 23.11 (2014), pp. 4750–4762.

damages the sample, leading to a hard limit on the number of projections that can be measured [MDG95]. In many other applications, there is a limit on the duration of a single scan. To decrease the scan duration, one can either acquire fewer projections or use a reduced dose per projection. In industrial tomography, process speed considerations limit the duration of each scan [Sip93].

Algebraic methods are based on a discrete representation of the tomographic problem, leading to a linear system of equations. This system is solved to obtain a reconstructed image. Since algebraic methods use a model of the actual data that is available, they usually yield more accurate reconstructions from limited data than analytical methods. Furthermore, by using specific ways of solving the linear system, it is possible to reduce the effect of noise on the reconstruction. An important drawback of algebraic methods is that they are computationally more expensive than analytical methods. The linear system that has to be solved is usually very large, and the iterative methods that are used often need a large number of iterations to converge to an acceptable solution.

In many applications of computed tomography, the computational efficiency of a reconstruction method is an important consideration. For example, in fast x-ray micro-tomographic experiments at synchrotrons, the speed of the post-processing pipeline has to match the high speed of data acquisition [Mok+13]. In fact, the computation efficiency of the FBP method is an important reason for why it is still commonly used instead of more advanced reconstruction methods [PSV09].

Methods that reduce the computation time of algebraic methods have been proposed by other authors. One approach is to implement algebraic methods more efficiently by using graphic processing units (GPUs) [XM05; Pan+11]. Other approaches focused on improving the convergence of algebraic methods, for example by improving the properties of the linear system [GB08]. Although these improvements reduce the computation time of algebraic methods significantly, even faster methods can be obtained by changing the algebraic methods themselves.

One such approach is taken in [BP12], where an angle-dependent FBP filter is calculated, such that the resulting FBP method approximates an algebraic method. Although the resulting method is able to approximate the algebraic method well, calculating the filter requires a large number of runs of the algebraic method, which is computationally expensive. The resulting filter can be reused for problems with identical projection geometry, but a change in geometry requires calculation of a new filter.

A filter that approximates an algebraic method is also derived in [Zen12], in which a reformulation of the SIRT algebraic method is translated to a fixed filter for the FBP method. An extension of the method for noisy projection data is given in [ZZ13]. The derived filter does not depend on the scanning geometry of the problem, and during derivation it is assumed that enough projections are available such that certain approximations are accurate. As such, the resulting method has more in common with analytical reconstruction methods than with algebraic methods.

A different approach, specific to tomosynthesis, is proposed in [Nie+12]. Instead of calculating a reconstruction image directly, Nielsen et al. calculate a filter *matrix*, which is multiplied with the projection data. The result is backprojected to produce the final reconstruction. Nielsen et al. show that their filter matrix can be formed efficiently in the case of tomosynthesis, but a complex method is needed to obtain this efficiency. Similar to [BP12], a change in geometry requires calculation of a new filter. Other methods for tomosynthesis use algebraic reconstructions of certain test objects to create filters [Kun+07; Lud+08].

In this chapter, we introduce a new reconstruction method, the *minimum residual filtered backprojection* method (MR-FBP), that combines ideas from both the analytical and algebraic approach, resulting in a method with a data-dependent filter. The method is based on an algebraic model of the tomographic problem, resulting in a method that can reconstruct problems with limited data more accurately than analytical methods. The linear system that has to be minimized, however, is based on filtered backprojection. Therefore, the system is much smaller than the ones used in algebraic methods or other approaches, making the method computationally efficient. Furthermore, we are able to use filtered backprojection to form our linear system, leading to a simple and efficient implementation.

This chapter is structured as follows. We formally define the tomographic reconstruction problem and analytical and algebraic reconstruction methods in Section 2.2. In Section 2.3, we introduce and explain the key contribution of this chapter: the minimum residual filtered backprojection method. Considerations concerning its implementation are discussed in Section 2.4. An extension of the method is given in Section 2.5, where additional constraints are added to its linear system to improve reconstruction quality. The experiments we performed to test the new method are explained in Section 2.6. Results, where we compare MR-FBP with popular reconstruction methods, are given in Section 2.7, along with a discussion on the interpretation of the results. Finally, we conclude the chapter in Section 2.8, where we give a summary and some final remarks.

2.2 Notation and concepts

In this section, we will explain the mathematical notation used throughout the chapter, and introduce all relevant concepts. We begin by formally defining the tomographic reconstruction problem. Filtered backprojection and algebraic methods are explained, and their mathematical definitions are given.

Problem definition

We consider the problem of reconstructing a two-dimensional object from its parallel-beam projections, with a single rotation axis. The approach we introduce here can be adapted to other geometries as well, such as fan-beam or cone-beam projections. The unknown object is modeled as a finite and integrable two-dimensional function $f : \mathbb{R}^2 \to \mathbb{R}$ with bounded support.

Define a line $l_{\theta,t}$ by its characteristic equation $t = x\cos\theta + y\sin\theta$. The line integral

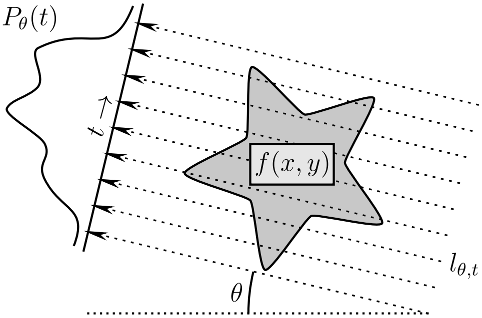


Figure 2.1: The two-dimensional tomography model used in this chapter. Parallel lines, rotated by angle θ, pass through the object f. A line $l_{\theta,t}$ has the characteristic equation $t = x\cos\theta + y\sin\theta$, and a projection $P_\theta(t)$ of f is given by the line integral of f over the line $l_{\theta,t}$.

$P_\theta(t)$ of f over a single line $l_{\theta,t}$ is given by:

$$P_\theta(t) = \int_{l_{\theta,t}} f(x,y)\,\mathrm{d}s \tag{2.1}$$

$$= \iint_{\mathbb{R}^2} f(x,y)\delta(x\cos\theta + y\sin\theta - t)\,\mathrm{d}x\mathrm{d}y \tag{2.2}$$

The *tomographic reconstruction problem* is concerned with the reconstruction of the unknown object f from its measured projections $P_\theta(t)$ for different combinations of θ and t. This projection geometry is shown graphically in Fig. 2.1.

In practice, only a finite set of projections P_θ are measured, one for each combination of projection angle $\theta \in \Theta = \{\theta_0, \theta_1, \ldots, \theta_{N_\theta - 1}\}$ and detector $k \in \{0, 1, \ldots, N_d - 1\}$, where N_θ is the number of projection angles, and N_d the number of detectors. Relative to the central detector, the position of a detector k is given by τ_k:

$$\tau_k = s\left(k - \frac{N_d - 1}{2}\right), \tag{2.3}$$

where s is the width of a detector. The entire set of measured detector positions is given by $T = \{\tau_0, \tau_1, \ldots, \tau_{N_d - 1}\}$. Using the measured projection data, the unknown object is reconstructed on an $N \times N$ grid of square pixels. We assume, without loss of generality, that the width of each pixel is equal to 1. Often, the number of pixels in each row of the reconstruction grid is taken equal to the number of detectors.

Filtered backprojection

One approach to solving the tomographic reconstruction problem is to take Eq. (2.2), and try to find an expression for $f(x,y)$ from this equation. The *filtered backprojection* method (FBP) is a result of this approach, and starts with convolving the projections

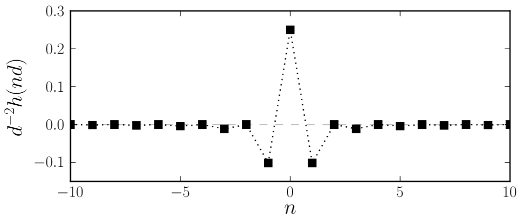


Figure 2.2: The Ram-Lak (ramp) filter for the FBP method, in real space. This filter is a discrete approximation of the optimal filter.

with a filter $h : \mathbb{R} \to \mathbb{R}$:

$$q_\theta(t) = \int_{-\infty}^{\infty} h(\tau) P_\theta(t-\tau)\mathrm{d}\tau \tag{2.4}$$

This convolution can be also be performed in Fourier space, where $\hat{P}$ and $\hat{h}$ denote the Fourier transforms of P and h respectively:

$$q_\theta(t) = \int_{-\infty}^{\infty} \hat{h}(u)\hat{P}_\theta(u)e^{2\pi \imath ut}\mathrm{d}u \tag{2.5}$$

One can show [KS01] that we obtain an expression for $f(x,y)$ if we take $\hat{h}(u) = |u|$:

$$f(x,y) = \int_0^{\pi} q_\theta(x\cos\theta + y\sin\theta)\mathrm{d}\theta \tag{2.6}$$

In practice, it is not possible to use Eq. (2.6) to reconstruct the object, since it requires $P_\theta(t)$ to be known for *all* angles $\theta \in [0,\pi)$ and $t \in \mathbb{R}$. Instead, we only know $P_\theta(t)$ for the measured angles Θ and detector positions T. To be able to use these discrete measurements, Eq. (2.6) has to be discretized, after which the filtered backprojection method is obtained:

$$f(x,y) \approx FBP_h(x,y) = \sum_{\theta_d\in\Theta}\sum_{\tau_p\in T} h(\tau_p)P_{\theta_d}(t-\tau_p) \tag{2.7}$$

where $t = x\cos\theta_d + y\sin\theta_d$. Since $t-\tau_p$ is usually not equal to one of the measured detector positions, some interpolation is needed to find the value of $P_{\theta_d}(t-\tau_p)$. Linear interpolation is often used, since projection data is usually reasonably smooth.

The filter h is only needed for discrete positions $\tau_p \in T$, and is therefore usually specified as a vector $\boldsymbol{h}$. Several discrete approximations of the optimal filter $\hat{h}(u) = |u|$ are used in practice, such as the Ram-Lak (ramp), Shepp-Logan, and Hann filters [Far+97; WWH05]. One of the most popular filters is the Ram-Lak filter, where we take the optimal $\hat{h}(u) = |u|$, and set $\hat{h}(u) = 0$ when $u > u_c$ for some u_c. This filter is shown, in real space, in Fig. 2.2.

The filtered backprojection method can be interpreted as a two-step process. First, the projection data is filtered by convolving it with filter $\boldsymbol{h}$. Afterwards, the result is

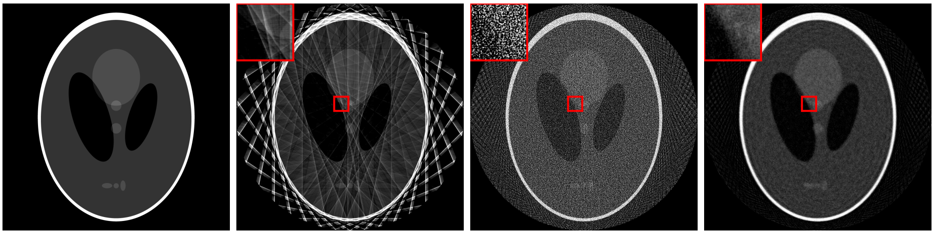


Figure 2.3: Three reconstructions of the Shepp-Logan head phantom (a), showing artifacts that occur with imperfect data. In (b) and (c), FBP was used to reconstruct the images. In (b), data from only 16 projections were used, and severe streak artifacts are present in the result. In (c), data from 1024 projections were used, but a large amount of Poisson noise was present, resulting in severe noise in the resulting image. In (d), SIRT, an algebraic method, was used to reconstruct the image using the same noisy projection data as in (c). The algebraic reconstruction image (d) has less noise compared to the FBP reconstruction (c).

backprojected to obtain the reconstructed image. One of the advantages of FBP is that it is fast compared to other methods: the filtering step can be performed efficiently in Fourier space in $\mathcal{O}(N_\theta N_d \log N_d)$ time, and only one backprojection is needed, which can be performed in $\mathcal{O}(N_\theta N^2)$ time.

The quality of an FBP reconstruction depends on how well the discretized equation Eq. (2.7) approximates the continuous equation Eq. (2.6). If data for many projection angles (say, several hundred) are known, an FBP reconstruction is often highly accurate. However, when the number of projections is small compared to the size of the image, the approximation is not very accurate, and severe artifacts can appear in the reconstructed image. Furthermore, noise in the projection data can cause artifacts in the reconstruction as well. FBP with the Ram-Lak filter is especially sensitive to noise, since high-frequency components of the projection data are amplified by the filter. The artifacts can make subsequent analysis of the reconstruction very difficult. Examples of artifacts in FBP reconstructions of imperfect data are shown in Fig. 2.3.

Algebraic methods

A different approach to solving the tomographic problem is to use a discrete representation of the problem. Here, we represent the discrete projection data as a single vector $\boldsymbol{p} \in \mathbb{R}^{N_\theta N_d}$, and represent the unknown image as a vector $\boldsymbol{x} \in \mathbb{R}^{N^2}$. The *projection matrix* $\boldsymbol{W}$ has $N_\theta N_d$ rows and N^2 columns, with element w_{ij} specifying the contribution of pixel j to detector i. We refer to the product of $\boldsymbol{W}$ with an image $\boldsymbol{x}$ as the *forward projection* of $\boldsymbol{x}$. Similarly, the product of $\boldsymbol{W}^T$ with projection data $\boldsymbol{p}$ is referred to as the *backprojection* of $\boldsymbol{p}$. If we look at the definition of the discrete FBP method (Eq. (2.7)), we see that the backprojection in the FBP method is identical to multiplication of the filtered sinogram with $\boldsymbol{W}^T$.

Algebraic methods are usually designed to minimize the difference between the measured projection data $\boldsymbol{p}$ and the forward projection of the reconstruction image, $\boldsymbol{W}\boldsymbol{x}$, with respect to a certain vector norm. In the case of the ℓ_2-norm, the algebraic solution image $\boldsymbol{x}_{alg}$ is defined as:

$$\boldsymbol{x}_{alg} = \operatorname*{argmin}_{\boldsymbol{x}} ||\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}||_2^2 \tag{2.8}$$

The algebraic solution $\boldsymbol{x}_{alg}$ can be found by solving the linear system $\boldsymbol{W}\boldsymbol{x} = p$ in a least squares sense.

The algebraic linear system is typically too large to be solved directly. Therefore, an iterative optimization method is normally used, which can often exploit the sparse structure of the projection matrix to improve computational and memory requirements. Different iterative methods can be used, leading to various algebraic reconstruction methods. One example is SIRT [KS01], belonging to the class of Landweber iteration methods [Lan51], which uses a specific Krylov subspace method to minimize the projection error. A different method is CGLS [Bjö96], which uses a conjugate gradient method.

The advantage of using an algebraic method compared to analytical methods is that the projection matrix $\boldsymbol{W}$ can be adapted to the actual geometry that was used during scanning. Therefore, these methods use a model of the actual data that is available, instead of assuming perfect data, as in analytical methods. Another advantage of algebraic methods is that additional constraints can be imposed on the reconstructed image $\boldsymbol{x}$, which can be used to improve reconstructions by exploiting prior knowledge. For example, total variation minimization based methods use algebraic methods to minimize the projection error, with an additional constraint that the ℓ_1-norm of the gradient of $\boldsymbol{x}$ should be minimal as well [SP08].

The main disadvantage of algebraic methods compared to analytical methods is their computation time. Because of the large system size, and the number of iterations that are needed to solve them, the time to reconstruct an image using an algebraic method is often several orders of magnitude larger than filtered backprojection, even when optimized for graphic processor units (GPUs) [XM05]. In the next section, we introduce a new reconstruction method that uses ideas from algebraic methods to improve filtered backprojection, leading to a method that is both fast and accurate.

2.3 Minimum residual filtered backprojection

We will now present the key contribution of this chapter: the minimum residual filtered backprojection method (MR-FBP). We start by noting that the FBP method is a linear operation on the projection data. In other words, the operation of the FBP algorithm can be modeled as a linear operator $\mathscr{M} : \mathbb{R}^{N_d N_\theta} \rightarrow \mathbb{R}^{N^2}$ applied to the projection data $\boldsymbol{p}$, which can be written as a $N^2 \times N_d N_\theta$ matrix $\boldsymbol{M_h}$:

$$FBP_{\boldsymbol{h}}(\boldsymbol{p}) = \boldsymbol{M_h}\boldsymbol{p} \tag{2.9}$$

As explained in Section 2.2, FBP consists of a convolution of $\boldsymbol{p}$ with filter $\boldsymbol{h}$, followed by a backprojection of the result:

$$\boldsymbol{M_h}\boldsymbol{p} = \boldsymbol{W}^T\boldsymbol{C_h}\boldsymbol{p} \tag{2.10}$$

where $\boldsymbol{C}_{\boldsymbol{h}}\boldsymbol{p}$ is the convolution of $\boldsymbol{p}$ by $\boldsymbol{h}$, specified by the $N_d N_\theta \times N_d N_\theta$ matrix $\boldsymbol{C}_{\boldsymbol{h}}$.

One of the properties of a convolution of two vectors is that it is *commutative*. Therefore, we can exchange the positions of $\boldsymbol{h}$ and $\boldsymbol{p}$ in Eq. (2.10):

$$\boldsymbol{M}_{\boldsymbol{h}}\boldsymbol{p} = \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h} \tag{2.11}$$

Up to this point, we have only rewritten the equation of the FBP method. Now, we will improve the method by changing the filter $\boldsymbol{h}$ from one of the standard filters to one specific to the problem we are solving. To calculate the specific data-dependent filter $\boldsymbol{h}^*$, we minimize the squared difference of the projections of the reconstruction with the measured projection data, similar to algebraic methods:

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}}{\operatorname{argmin}} \|\boldsymbol{p} - \boldsymbol{W}\ FBP_{\boldsymbol{h}}(\boldsymbol{p})\|_2^2 \tag{2.12}$$

Using Eqs. (2.9) and (2.11), we can write this as:

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}}{\operatorname{argmin}} \left\|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{h}\right\|_2^2 \tag{2.13}$$

As with the algebraic methods, we can find $\boldsymbol{h}^*$ by solving the following linear system for $\boldsymbol{h}$ in the least squares sense:

$$\boldsymbol{A}_{\boldsymbol{p}}\boldsymbol{h} = \boldsymbol{p} \tag{2.14}$$

where $\boldsymbol{A}_{\boldsymbol{p}} = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}$.

After computing the least squares solution $\boldsymbol{h}^*$ to the linear system of Eq. (2.14), the MR-FBP reconstruction is obtained by computing the FBP reconstruction with $\boldsymbol{h}^*$ as filter. The complete MR-FBP algorithm is given by:

Algorithm 2.1 MR-FBP reconstruction method

```
1. Calculate $A_p = W W^T C_p$.
2. Find least squares solution $h^*$ of $A_p h = p$.
3. Return $FBP_{h^*}(p)$ as MR-FBP reconstruction.
```

The linear system we need to solve in step 2) looks similar to the system $\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}$, which is solved in the least squares sense by algebraic methods (Eq. (2.8)). The difference is that the system of Eq. (2.14) has fewer unknowns: $\boldsymbol{A}_{\boldsymbol{p}}$ has N_d columns if we impose that $\boldsymbol{h}$ is angle-independent, while $\boldsymbol{W}$ has N^2 columns. As we will show in Section 2.4, we are able to reduce the number of columns of $\boldsymbol{A}_{\boldsymbol{p}}$ to $\mathcal{O}(\log N_d)$ by exponential binning, without reducing the reconstruction quality significantly.

Because of the large size of the linear system that needs to be solved in Eq. (2.8), algebraic methods usually use an iterative method to find least squares solutions. These iterative methods can sometimes converge slowly, and they introduce a new parameter to the method: the number of iterations to perform. Since the system of MR-FBP is smaller, we can use a direct method to find the least squares solution, making it both efficient and parameter-free.

2.4 Implementation

Although the number of unknowns of the MR-FBP method is smaller than that of algebraic methods, the actual implementation of the method is important to actually obtain a method that is computationally more efficient. In this section, we give details on how we implemented the MR-FBP method in this chapter to obtain the experimental results of Section 2.7. We will begin by discussing how the matrix $\boldsymbol{A_p}$ of Eq. (2.14) can be calculated efficiently. Furthermore, we will show that the size of the linear system can be reduced by exponential binning. Finally, we discuss the computational complexity of the MR-FBP method compared to existing methods.

Calculation of A_p

The first step of the MR-FBP method is to calculate the matrix $\boldsymbol{A_p} = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C_p}$. Usually, the projection matrix $\boldsymbol{W}$ is not used directly by algebraic methods, since it can be very large. Instead, multiplication of $\boldsymbol{W}$ with an image $\boldsymbol{x}$ is calculated implicitly by calculating the line integrals of $\boldsymbol{x}$ on-the-fly [PBS11]. Similarly, multiplication of $\boldsymbol{W}^T$ with a sinogram $\boldsymbol{p}$ is calculated by backprojecting $\boldsymbol{p}$ on-the-fly. Here, we use a similar approach to calculate $\boldsymbol{A_p}$, column by column.

Denoting a column j of $\boldsymbol{A_p}$ by $\boldsymbol{A_p}(:,j)$, we have:

$$\boldsymbol{A_p}(:,j) = \boldsymbol{A_p}\boldsymbol{e}_j \tag{2.15}$$

where $\boldsymbol{e}_j$ is a unit vector with all elements zero except for element j, which is equal to one. Using the definition of $\boldsymbol{A_p}$, we see that:

$$\boldsymbol{A_p}(:,j) = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C_p}\boldsymbol{e}_j = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{e}_j}\boldsymbol{p} = \boldsymbol{W}\ \mathit{FBP}_{\boldsymbol{e}_j}(\boldsymbol{p}) \tag{2.16}$$

In other words, we can calculate a column j of $\boldsymbol{A_p}$ by creating a filtered backprojection reconstruction with filter $\boldsymbol{e}_j$, and forward projecting the result. By doing this for every column, we can calculate matrix $\boldsymbol{A_p}$.

Exponential binning

At this point, the MR-FBP linear system of Eq. (2.14) has $N_d N_\theta$ equations and N_d unknowns, one for each detector element. Although the system is smaller than the one used in algebraic reconstruction methods, which have $N_d N_\theta$ equations and N^2 unknowns, we can further reduce the number of unknowns by *exponential binning*. Exponential binning was also used successfully to reduce system sizes in [BK06] and Chapter 5 of this thesis.

In exponential binning, we assume that the filter $\boldsymbol{h}$ is a piecewise constant function of N_b pieces. Each constant region of the function is called a *bin*, and the boundary points of a bin β_i are defined by positions s_i and s_{i+1}: $\beta_i = (s_i, s_{i+1})$. The width of a bin is equal to the difference between its boundary points $d_i = s_{i+1} - s_i$. Since the filter value of a single bin is constant, we can represent a binned filter by a vector with

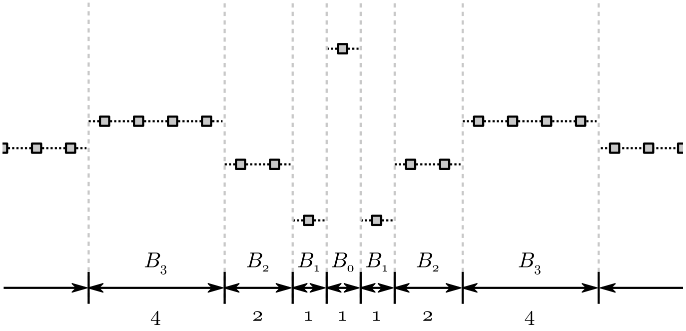


Figure 2.4: A filter with exponential binning, where we impose that the filter is symmetrical, and consists of several bins with a constant filter value. The size of the bins increases exponentially away from the center of the filter.

one element for each bin. The idea of exponential binning in the MR-FBP method is that we can reduce the number of unknowns of the linear system from N_d to N_b, by using fewer bins than detectors ($N_b < N_d$). The question remains how to choose the boundary points of the bins.

Looking at Fig. 2.2, we see that the Ram-Lak filter has most details around $n = 0$, and drops to zero relatively quickly for $|n| \rightarrow \infty$. This suggests that we should use small bins around $n = 0$, and can use larger bins further away from the center. In this chapter, we use bins with widths that increase exponentially away from $n = 0$. Specifically, we take $d_i = 1$ for $|i| < N_l$ and $d_i = 2^{|i|-N_l}$ for $|i| \geq N_l$, with β_0 being the central bin. The number of bins with width 1 is specified by N_l, where larger values lead to more detail around the center of the filter, but more unknowns as well. For the rest of this chapter, we used $N_l = 2$, unless specified otherwise. We can reduce the number of bins even more by making it symmetric, defining new bins $B_0 = \beta_0$ and $B_i = (\beta_i \cup \beta_{-i})$ for $i \neq 0$. A filter with exponential binning and $N_l = 2$ is shown in Fig. 2.4.

Since the bin width increases exponentially, we end up with $\mathcal{O}(\log N_d)$ bins. Therefore, by using exponential binning, we have reduced the number of unknowns of the MR-FBP method from N_d to $\mathcal{O}(\log N_d)$. The restrictions we impose on the filter by assuming it is piecewise constant and symmetrical can reduce the quality of the MR-FBP reconstructions. We will show in Section 2.7, however, that the quality does not decrease significantly by using exponential binning, while the time it takes to calculate the reconstructions greatly decreases.

The matrix $\boldsymbol{A_p}$ with an exponentially binned filter can again be calculated column by column. In order to do this, we change the filter $\boldsymbol{e}_j$ of Eq. (2.16) to a vector $\boldsymbol{q}_{B_i}$, in which each filter element included in bin B_i is set to one, and all other elements are set to zero.

Computational complexity

For many tomographic reconstruction methods, the most costly subroutines computationally are forward projecting and backprojecting, for which straightforward implementations take $\mathcal{O}(N_d N_\theta N)$ and $\mathcal{O}(N_\theta N^2)$ time, respectively, although faster implementations exist which use hierarchical decomposition [BB00]. We can compare the computational costs of different reconstruction methods by comparing the number of projection operations each method has to perform. Filtered backprojection consists of a single projection operation: the final backprojection of the filtered sinogram. Algebraic methods usually perform a few projection operations per iteration. The SIRT method, for example, performs two projection operations per iteration, and typically has to perform $\mathcal{O}(N_d)$ iterations to converge to an acceptable solution.

The MR-FBP method has to perform one forward projection and one backprojection for every column of $\boldsymbol{A_p}$ during its calculation. Because there are $\mathcal{O}(\log N_d)$ unknowns, MR-FBP has to perform $\mathcal{O}(\log N_d)$ projections. The total computation time of calculating $\boldsymbol{A_p}$ becomes $\mathcal{O}\left((N_\theta N_d N + N_\theta N^2)\log N_d\right)$. If we assume that $N_d \approx N$, which is often the case, the total computation time becomes $\mathcal{O}\left(N_\theta N_d^2 \log N_d\right)$. To summarize, FBP, SIRT, and MR-FBP have to perform $\mathcal{O}(1)$, $\mathcal{O}(N_d)$, and $\mathcal{O}(\log N_d)$ projections, respectively, which shows that MR-FBP has to perform significantly fewer operations than SIRT.

Of course, the MR-FBP method also has to find the least squares solution to its linear system of Eq. (2.14). Because of its smaller size however, we can use direct methods to find this solution, instead of the iterative methods used in algebraic methods. The direct method we used in this chapter to generate the results of Section 2.7, the `gels*` lapack routine, uses singular value decomposition, and can solve an $m \times n$ system in $\mathcal{O}(mn^2)$ time. Since the MR-FBP system has $N_d N_\theta$ equations and $\log N_d$ unknowns, the least squares filter $\boldsymbol{h}^*$ can be found in $\mathcal{O}(N_d N_\theta [\log N_d]^2)$ time. Summing both the calculation of $\boldsymbol{A_p}$ and of $\boldsymbol{h}^*$, the total computation time of the MR-FBP method becomes $\mathcal{O}\left(N_\theta N^2 \log N_d + N_d N_\theta [\log N_d]^2\right)$.

2.5 Additional constraints

The reconstruction quality of algebraic reconstruction methods can be improved by exploiting prior knowledge about the object that was scanned. One approach of exploiting this knowledge is to add an additional constraint to the system that is minimized. Formally, such a reconstruction $\boldsymbol{x}^*$ can be found by solving the following equation:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[\|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 + \lambda g(\boldsymbol{x}) \right] \tag{2.17}$$

where $g(\boldsymbol{x})$ is a function depending on the type of prior knowledge that is exploited. For example, if one knows that the object that is reconstructed has a sparse gradient, total-variation minimization can be used by setting $g(\boldsymbol{x}) = \|\nabla \boldsymbol{x}\|_1$ [SP08]. The parameter λ controls the relative strength of the additional constraint compared to the data fidelity term $\|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2$. The optimal value of λ is often difficult to find, as it depends on the scanned object and acquired projection data.

A similar approach can be applied to the MR-FBP method, by imposing an additional constraint on the optimal filter $\boldsymbol{h}^*$:

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}}{\operatorname{argmin}} \left[\left\| \boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h} \right\|_2^2 + \lambda g(\boldsymbol{h}) \right] \tag{2.18}$$

Different functions f can be used to exploit various kinds of prior knowledge. In this chapter, we will use one example, where the change in intensity of the reconstructed image in the horizontal direction and vertical direction is minimized. This can be achieved by letting $g(\boldsymbol{h}) = \|\nabla_x \boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h}\|_2 + \|\nabla_y \boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h}\|_2$, where $\nabla_x \boldsymbol{f}$ denotes the horizontal gradient of image $\boldsymbol{f}$, and $\nabla_y \boldsymbol{f}$ the vertical gradient. The horizontal and vertical gradient can be approximated by the linear Sobel operators $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$, which convolve the image with two-dimensional kernels $\boldsymbol{G}_x$ and $\boldsymbol{G}_y$, respectively:

$$\boldsymbol{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, \boldsymbol{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.19}$$

If we approximate the gradients by $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$, we can add the additional constraints to the linear MR-FBP system, as additional equations:

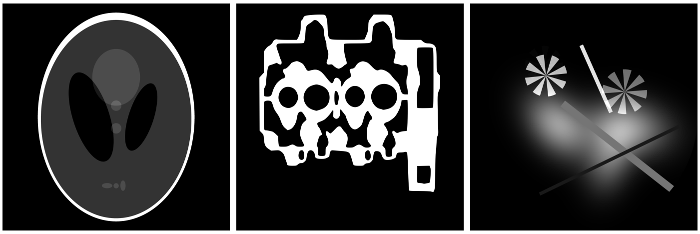
$$\begin{bmatrix} \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}} \\ \lambda\boldsymbol{D}_x\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}} \\ \lambda\boldsymbol{D}_y\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}} \end{bmatrix} \boldsymbol{h} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{2.20}$$

The least squares solution $\boldsymbol{h}^*_{GM}$ of this system can be found using standard methods, by solving:

$$\boldsymbol{h}^*_{GM} = \underset{\boldsymbol{h}}{\operatorname{argmin}} \left[\left\| \boldsymbol{p} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h} \right\|_2^2 + \lambda \left(\left\| \boldsymbol{D}_x\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h} \right\|_2^2 + \left\| \boldsymbol{D}_y\boldsymbol{W}^T\boldsymbol{C}_{\boldsymbol{p}}\boldsymbol{h} \right\|_2^2 \right) \right] \tag{2.21}$$

The resulting method, which we call MR-FBP$_{\mathrm{GM}}$, finds a filter that minimizes a weighted sum of the residual and the horizontal and vertical gradient of the resulting reconstruction. The method can improve reconstructions of objects that have a small gradient. In the case of noise in the projection data, MR-FBP$_{\mathrm{GM}}$ can improve reconstructions as well, since the gradient of the object is usually much smaller than that of image noise. Therefore, by reducing the gradient of the reconstructed image, we reduce the amount of image noise as well.

Similar to the MR-FBP method, we can calculate the matrix $\boldsymbol{A}_{\boldsymbol{p}}$ of the MR-FBP$_{\mathrm{GM}}$ method column by column. For a column j, we can calculate the FBP reconstruction with filter $\boldsymbol{e}_j$. We can then forward project this reconstruction to obtain the top part of column j of the linear system shown in Eq. (2.20). The remaining part of column j can be calculated by applying the Sobel operators $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ to the FBP reconstruction. Since the gradient image calculations can be performed efficiently in Fourier space, the asymptotic computational complexity of the $\boldsymbol{A}_{\boldsymbol{p}}$ calculation step of MR-FBP$_{\mathrm{GM}}$ is identical to MR-FBP. The resulting linear system consists of $N_d N_\theta + 2N^2$ equations and $\log N_d$ unknowns. Therefore, the linear system can be solved in



(a) PHANTOM1 (b) PHANTOM2 (c) PHANTOM3

Figure 2.5: The three phantom images used in this chapter. PHANTOM1 is the Shepp-Logan head phantom, PHANTOM2 represents a cross section of an engine block, and PHANTOM3 is a difficult to reconstruct object with both discrete and continuous areas.

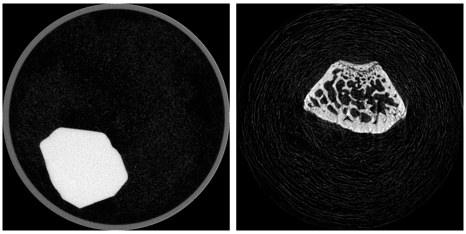
$\mathcal{O}((N_d N_\theta + N^2)[\log N_d]^2)$ time, which is a slightly higher complexity than MR-FBP without gradient minimization. The total computational complexity of MR-FBP$_{\mathrm{GM}}$ is equal to $\mathcal{O}\left(N_\theta N^2 \log N_d + (N_d N_\theta + N^2)[\log N_d]^2\right)$.

2.6 Experiments

To compare the performance of the MR-FBP and MR-FBP$_{\mathrm{GM}}$ methods with other methods, we implemented them using Python 2.7.3, PyCUDA 2012.1, and Numpy 1.6.3 [Oli07] built with ATLAS 3.10.0 [WP05]. The GPU implementations of the forward and backprojection operations are based on the ASTRA-Toolbox [PBS11], in which backprojection is not the exact transpose of forward projection for performance reasons. We applied MR-FBP on three phantom images and two experimental μ-CT datasets, comparing the results of MR-FBP with SIRT, an algebraic method, and FBP with three different standard filters: the Ram-Lak filter, the Hann filter, and the Shepp-Logan filter. We compare MR-FBP$_{\mathrm{GM}}$ reconstructions with MR-FBP, FBP, and SIRT reconstructions of one of the three phantoms, with noise in the projection data.

The three phantom images are shown in Fig. 2.5. Each phantom image is represented on a 4096 × 4096 pixel grid, on which projections are calculated. Afterwards, the projection data is rebinned to 1024 detector elements, and all reconstructions are calculated on a 1024 × 1024 pixel grid. We calculate reconstructions for varying numbers of projection angles, and compare them to the original phantom image, scaled to a 1024 × 1024 pixel grid by averaging 4 × 4 squares.

For the experimental data, we used two different objects scanned by μCT scanners, with the acquired cone-beam projection data rebinned to a parallel beam geometry. The first object is a diamond, which was scanned by a Scanco 40 μCT scanner using 1024 detector elements and 500 equidistant projection angles. The second object is a mouse femur, scanned by a Skyscanner 1172 μCT scanner using 1200 detector elements and 360 equidistant projection angles. Reconstructions of the experimental data are shown in Fig. 2.6, calculated using SIRT and all projection angles. To test the reconstruction quality of MR-FBP compared to other methods, we apply them on projection data from small subsets of all projection angles, with a fixed separation between the angles.



(a) DIAMOND (b) FEMUR

Figure 2.6: Reconstructions of the μCT datasets used in this chapter. The reconstructions were calculated using SIRT and all available projections.

For each experiment, we report the mean absolute error and structural similarity (SSIM) index [Wan+04] of reconstructions of the various methods. The mean absolute error is defined as:

$$e_p(\boldsymbol{x},\boldsymbol{o}) = \frac{N_D^{-1} \sum_{i\in D} |\boldsymbol{x}_i - \boldsymbol{o}_i|}{\max \boldsymbol{o} - \min \boldsymbol{o}} \tag{2.22}$$

where $\boldsymbol{x} \in \mathbb{R}^{N^2}$ is the reconstructed image, $\boldsymbol{o} \in \mathbb{R}^{N^2}$ the correct image, and the average is taken over all N_D pixels within the central disc D of radius $N/2$. For the experimental data, the mean absolute errors and SSIM indices are calculated with respect to SIRT reconstructions from projection data of all available projections, shown in Fig. 2.6. The SSIM index measures the similarity between two images, and was designed to represent human visual perception more accurately than other metrics. A higher SSIM index corresponds with larger perceptual similarity between the compared images. For the phantom experiments, we also report the mean absolute residual, defined as:

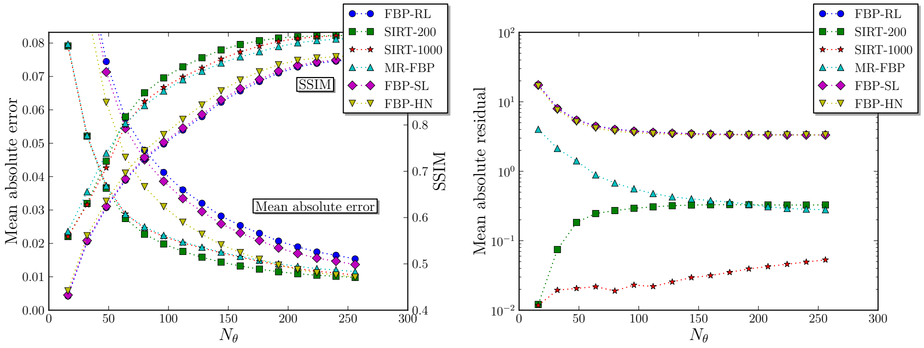
$$e_r(\boldsymbol{x},\boldsymbol{p}) = (N_d N_\theta)^{-1} \sum_{i=0}^{N_d N_\theta - 1} \left|(\boldsymbol{W}\boldsymbol{x})_i - \boldsymbol{p}_i\right| \tag{2.23}$$

where $\boldsymbol{x} \in \mathbb{R}^{N^2}$ is the reconstructed image, and $\boldsymbol{p} \in \mathbb{R}^{N_\theta N_d}$ the measured projection data.

2.7 Results and discussion

Results for simulation phantoms

The mean absolute error, SSIM index, and mean absolute residual for PHANTOM1 are shown in Fig. 2.7 as a function of the number of projection angles N_θ. The results show that the mean absolute error and mean absolute residual of the MR-FBP reconstructions are significantly lower than those of all tested FBP methods. Even though FBP with the Shepp-Logan filter or Hann filter produces more accurate reconstructions than FBP with the Ram-Lak filter, MR-FBP is more accurate than all three. A similar result is



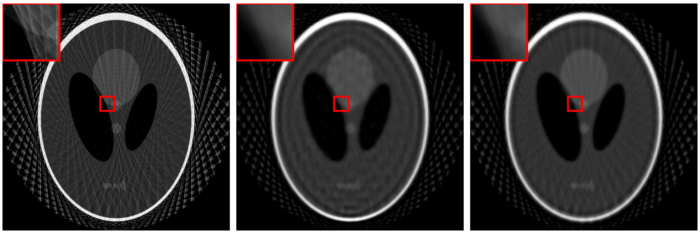
(a) (b)

Figure 2.7: The mean absolute error, SSIM, and mean absolute residual of reconstructions calculated with different methods, for PHANTOM1. The methods shown are FBP with the Ram-Lak filter (FBP-RL), FBP with the Shepp-Logan filter (FBP-SL), FBP with the Hann filter (FBP-HN), SIRT with 200 iterations (SIRT-200), SIRT with 1000 iterations (SIRT-1000), and the MR-FBP method (MR-FBP).

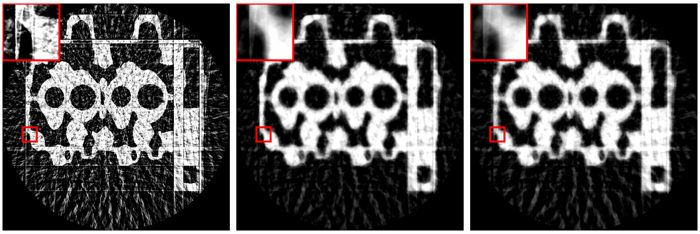
found for the SSIM, with significantly higher indices for SIRT and MR-FBP, compared to all tested FBP methods. Compared to SIRT, MR-FBP produces reconstructions with slightly higher errors, lower SSIM indices, and higher residuals. Later results in Fig. 2.9 will show, however, that MR-FBP is significantly faster than SIRT at producing these reconstructions. Results for the other two phantom images are similar to those of PHANTOM1.

For all three phantoms, reconstructions of FBP with the Ram-Lak filter, SIRT, and MR-FBP are shown in Fig. 2.8 for 32 projection angles. Note that in all comparison images in this chapter, the pixel value that a certain graylevel represents is identical for all compared methods. A zoomed inset is included in most images, giving a better indication than the entire image of how the reconstruction will look at full resolution. Figure 2.8 shows that both MR-FBP and SIRT are able to reduce the number of streak artifacts compared to standard FBP. Visually, the sharpness of the MR-FBP and SIRT reconstructions is slightly lower than that of the FBP reconstructions. In some applications, the higher sharpness of the FBP reconstructions might be preferable despite its artifacts, especially when the user is familiar with the scanned objects and FBP artifacts. In other applications, and in common post-processing steps such as segmentation, the artifacts present in FBP reconstructions can be problematic, and MR-FBP might be preferable.

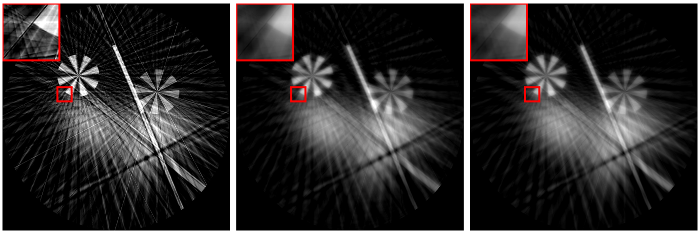
The time it takes to calculate the reconstructions of PHANTOM1 using the different methods is shown in Fig. 2.9. In Fig. 2.9a, the reconstruction time is shown as a function of the number of projections, for a fixed number of detectors $N_d = 1024$. The results show that MR-FBP is significantly faster than SIRT, but slower than FBP with a standard filter. Specifically, MR-FBP is around 20 times faster than SIRT with 200 iterations in these cases, which is expected since MR-FBP has to perform around $2 \log N_d = 20$ forward projections and backprojections, while SIRT with 200 iterations has to perform 400. Similar results are shown in Fig. 2.9b, where the reconstruction



(a) FBP-RL (b) SIRT-200 (c) MR-FBP

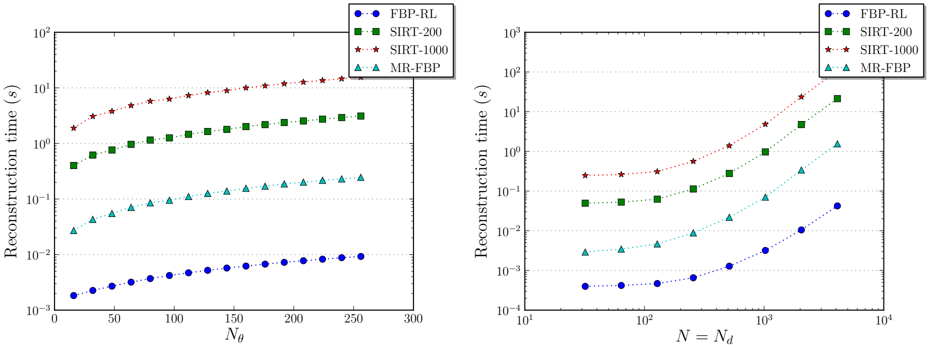


(d) FBP-RL (e) SIRT-200 (f) MR-FBP



(g) FBP-RL (h) SIRT-200 (i) MR-FBP

Figure 2.8: Reconstructions of the phantom images, from data with 32 projections, for different reconstruction methods.



(a) (b)

Figure 2.9: The reconstruction time of the PHANTOM1 image. In (a), $N = N_d = 1024$, and the reconstruction time is shown as a function of the number of projections N_θ. In (b), the number of projections is 64, and the reconstruction time is shown as a function of the number of detectors $N = N_d$.

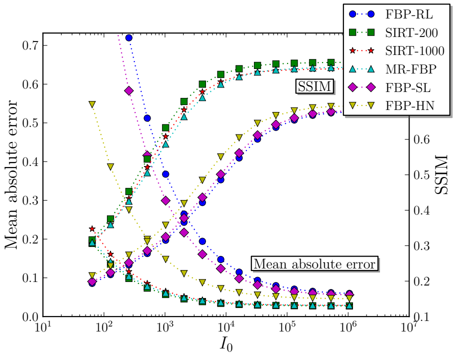


Figure 2.10: Mean absolute error and SSIM of reconstructions of PHANTOM1 from data of 64 projections with various amounts of Poisson noise applied.

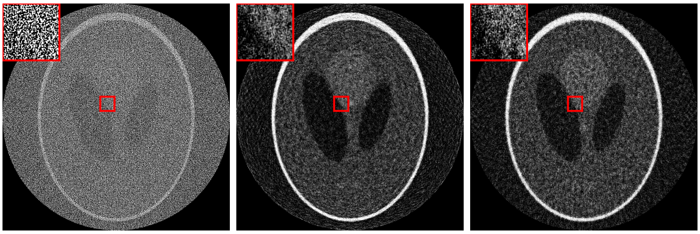
time is shown as a function of the number of detectors, for a fixed number of projections $N_\theta = 64$.

In Fig. 2.10, the mean absolute error of reconstructions of PHANTOM1 with different methods is shown, for data of 64 projections with various amounts of Poisson noise applied. The parameter I_0 indicates the amount of applied Poisson noise, with lower values corresponding to higher amounts of noise. Specifically, noise was applied by first transforming the projection data to virtual photon counts, where I_0 corresponds to the largest photon count of all detector elements. For each detector element, a new photon count is sampled from a Poisson distribution with the original photon count as expected value. The resulting noisy photon counts are transformed back to obtain noisy projection data.

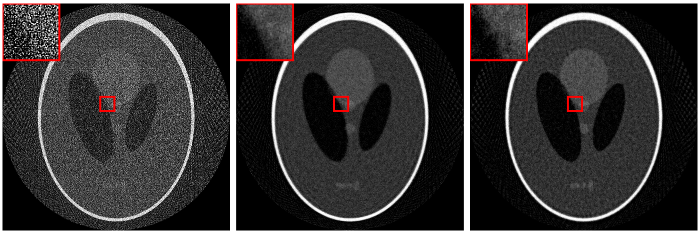
Since the Ram-Lak filter amplifies high-frequency signals, the reconstructions of FBP with the Ram-Lak filter are of low quality when noise is present in the projection data. Other filters, like the Hann filter, suppress high-frequency signals, and therefore yield reconstructions of higher quality. Algebraic techniques, like SIRT, often include a form of regularization on the reconstruction image, yielding reconstructions of even higher quality when noise is present. The results of Fig. 2.10 show that, as expected, SIRT reconstructions have the lowest error and highest SSIM, while the FBP method with the Hann filter yields reconstructions with higher errors and lower SSIM, and FBP with the Ram-Lak filter produces reconstructions with the highest error and lowest SSIM. The MR-FBP method yields reconstructions with similar errors and SSIM indices to the SIRT method, for every noise level, but requires less computation time. Examples of reconstructions of data with two different noise levels are shown in Fig. 2.11, for FBP with the Ram-Lak filter, SIRT, and the MR-FBP method.

Results for experimental data

The mean absolute error and SSIM indices of the reconstructions of the experimental data, obtained by the different methods, is shown in Fig. 2.12. For the DIAMOND

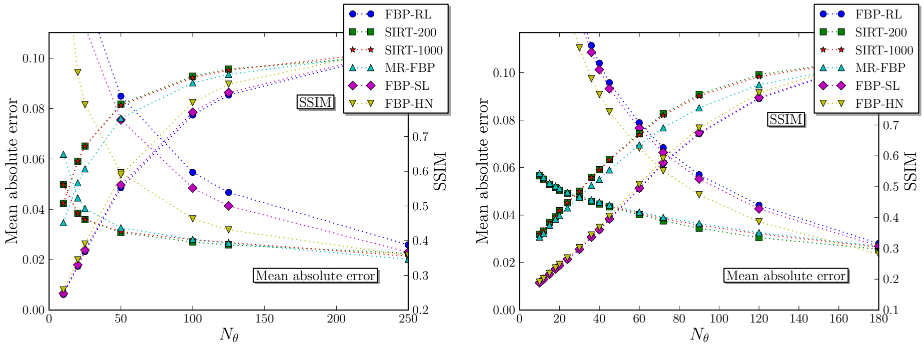


(a) FBP-RL, $I_0 = 2^6$ (b) SIRT-200, $I_0 = 2^6$ (c) MR-FBP, $I_0 = 2^6$



(d) FBP-RL, $I_0 = 2^{10}$ (e) SIRT-200, $I_0 = 2^{10}$ (f) MR-FBP, $I_0 = 2^{10}$

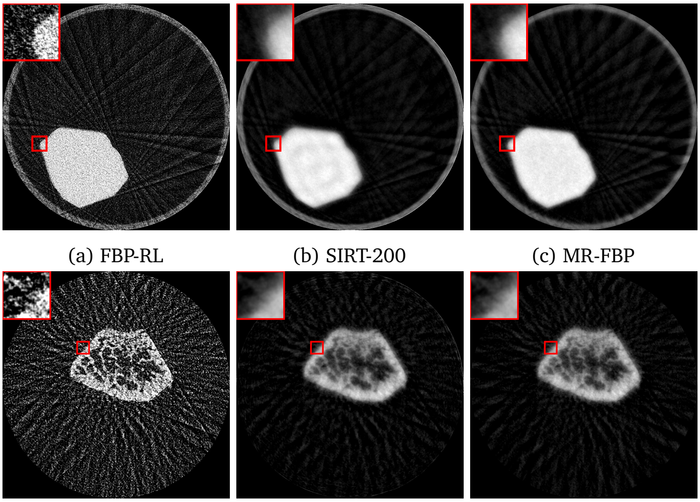
Figure 2.11: Reconstruction images of PHANTOM1, for FBP, SIRT-200, and MR-FBP, with various amounts of Poisson noise applied to data from 64 projections.



(a) DIAMOND (b) FEMUR

Figure 2.12: The mean absolute error and SSIM of reconstructions of the experimental data, for different numbers of projections.

dataset, the mean absolute error of the MR-FBP method is slightly higher than SIRT when reconstructing from very few projections, but both are significantly lower than FBP with fixed filters. For the FEMUR dataset, the mean absolute error of the MR-FBP method is similar to that of SIRT, with the error of FBP again significantly higher. Similar results can be observed for the SSIM indices of the different methods for both experimental sets. Reconstructions of both experimental datasets, with data from 25 projections for the DIAMOND set, and 24 projections for the FEMUR set, are shown in Fig. 2.13. Similar to previous results, the reconstructions of SIRT and MR-FBP are


(a) FBP-RL (b) SIRT-200 (c) MR-FBP

(d) FBP-RL (e) SIRT-200 (f) MR-FBP

Figure 2.13: Reconstructed images of both experimental datasets, for data of 25 projections (DIAMOND) and 24 projections (FEMUR).

visually similar, both having less artifacts than the FBP reconstructions, but a lower sharpness as well.

Additional constraints

To test the reconstruction quality of the MR-FBP$_{\mathrm{GM}}$ method, we used the PHANTOM2 phantom image, which has a sparse gradient, and calculated the mean absolute error of reconstructions obtained by different methods from data of 64 projections and various amounts of applied Poisson noise. The weight λ in Eq. (2.20) of the MR-FBP$_{\mathrm{GM}}$ method was set to $27 + 1600/I_0$, which was experimentally verified to be a reasonable choice for this phantom. Results are shown in Fig. 2.14, where it is clear that by using MR-FBP$_{\mathrm{GM}}$, which exploits prior knowledge about the gradient of the image, we are able to obtain significantly lower mean absolute errors and significantly higher SSIM compared to other methods. The computation times of a single reconstruction using the FBP, SIRT-200, MR-FBP, and MR-FBP$_{\mathrm{GM}}$ methods were 3.02 ms, 970 ms, 68.5 ms, and 577 ms, respectively. Reconstructions of the PHANTOM2 image from data with a large amount of applied Poisson noise ($I_0 = 2^6$) are shown in Fig. 2.15.

Exponential binning

To investigate the influence of exponential binning on the reconstruction quality and computation time of the MR-FBP method, we calculated the mean absolute error and reconstruction time both with and without exponential binning, for 64 projections of PHANTOM1. With exponential binning, the mean absolute error was equal to 0.0287,

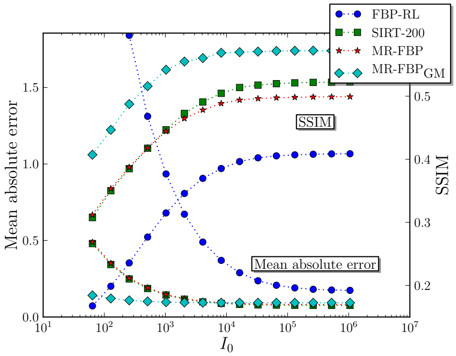
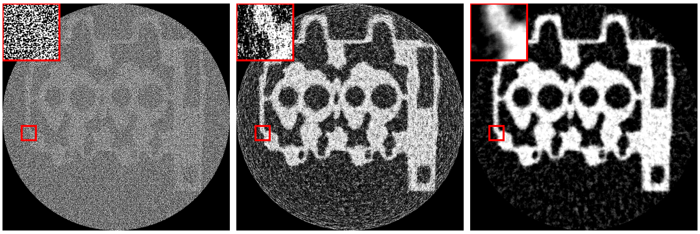


Figure 2.14: Mean absolute error and SSIM of reconstructions of PHANTOM2 from data of 64 projections with various amounts of Poisson noise applied. Shown are errors of FBP, SIRT-200, MR-FBP, and MR-FBP$_{GM}$.



(a) FBP-RL (b) SIRT-200 (c) MR-FBP$_{GM}$

Figure 2.15: Reconstructed images from 64 projections of PHANTOM2 with a large amount of applied Poisson noise ($I_0 = 2^6$).

and it took 0.0684 seconds to reconstruct the image. Without exponential binning, the mean absolute error increased slightly to 0.0289, while the computation time increased significantly to 6.1698 seconds. This result shows that by using exponential binning, the reconstruction quality of MR-FBP does not decrease significantly, while the reconstruction time is greatly improved. In fact, it seems that exponential binning has a slight regularizing effect on the reconstruction, since the mean absolute error is actually slightly smaller with exponential binning than without.

Computed filters

In Fig. 2.16, filters that are computed by the MR-FBP method (Eq. (2.13)) are shown in Fourier space for different reconstruction problems, with the standard Ram-Lak filter shown as well. A first observation is that the computed filters are significantly different from the standard Ram-Lak filter, except for low frequencies. A second observation is that the computed filters also differ significantly for different reconstruction problems. The computed filter for $N_\theta = 128$ is closer to the Ram-Lak filter than the one for $N_\theta = 64$, which is expected, since the Ram-Lak filter is ideal when an infinite number of projections is available. Furthermore, when noise is present in the projection data,

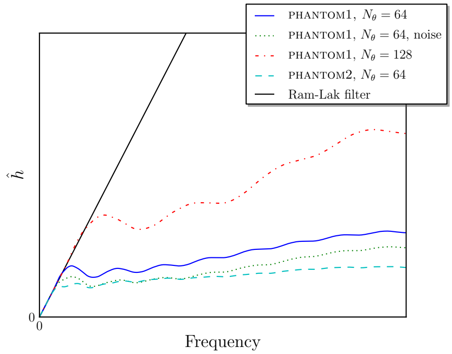


Figure 2.16: Filters computed by the MR-FBP method for different reconstruction problems ($N_d = N = 1024$) and the standard Ram-Lak filter, shown in Fourier space.

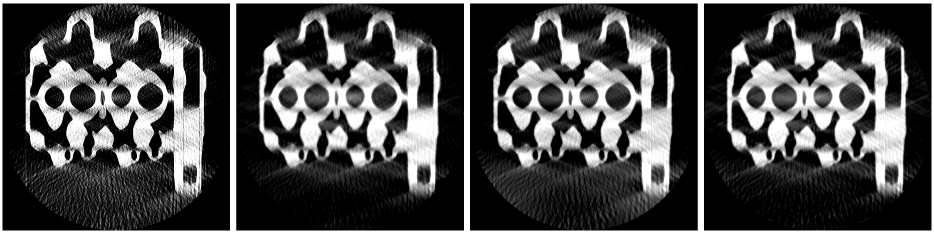
the computed filter has a lower response for high frequencies, filtering out some of the high-frequency noise. Finally, when reconstructing PHANTOM2 instead of PHANTOM1, the computed filter is different as well. These results show that the approach of calculating a filter specific to the reconstruction problem is valid, and that there is no single filter that is ideal for every problem.

Limited angular range

In some applications of computed tomography, it is impossible to acquire projections over the full $[0, \pi]$ angular range. Since analytical methods assume that data of the full angular range is available, a limited angular range can lead to severe artifacts in the reconstructed image. Algebraic methods include the limited angular range in their model, and can usually produce more accurate reconstructions than analytical methods. When presented with data with a limited angular range, reconstructions of the MR-FBP method contain similar artifacts to those reconstructed by standard FBP, as is visible in Fig. 2.17a and Fig. 2.17c. A possible cause for these artifacts is that the MR-FBP method uses an angle-independent filter: the projections of each angle are filtered by the same filter. Indeed, if we extend MR-FBP to use an angle-dependent filter, MR-FBP reconstructions are similar to those of algebraic methods, as shown in Fig. 2.17b and Fig. 2.17d. To use angle-dependent filters in MR-FBP, we increase the size of the unknown filter $\boldsymbol{h}$ in Eq. (2.14) such that it includes a different filter for each projection angle. Of course, the size of the resulting linear system is increased as well, and the method will be significantly slower than angle-independent MR-FBP.

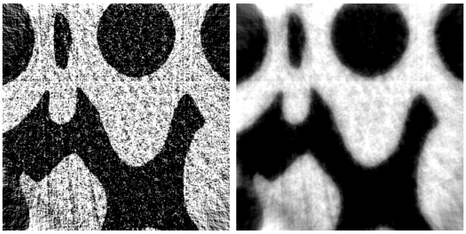
Truncated projection data

When the scanned object is larger than the field of view of the experimental setup, the acquired projection data is truncated at the edges of the detector. The MR-FBP method can be adapted to these cases by combining the standard approach of handling



(a) FBP-RL (b) SIRT-200 (c) MR-FBP (d) MR-FBP-θ

Figure 2.17: Reconstructions from 64 projections of PHANTOM2 with a limited angular range of $\frac{3}{4}\pi$. Reconstructions are shown for standard FBP, SIRT with 200 iterations, standard MR-FBP, and MR-FBP with an angle-dependent filter (MR-FBP-θ).



(a) FBP-RL (b) MR-FBP

Figure 2.18: Reconstruction images of PHANTOM2, for FBP and MR-FBP, from projection data of 64 projections truncated to 256 detector elements with applied Poisson noise.

truncated data in the FBP method with the standard approach in algebraic methods. First, the projection data has to be padded with a smoothly decreasing function in order to avoid the sharp drop to zero at the edge of the detector, similar to the standard approach for FBP. Secondly, during calculation of the MR-FBP matrix, we backproject the padded projection data onto a reconstruction grid as wide as the padded detector, but forward project onto the unpadded detector, similar to the standard approach for iterative methods. By using this combined approach, the MR-FBP method is able to reconstruct truncated projection data. An example of a reconstruction from truncated projection data is shown in Fig. 2.18, where we artificially truncated projection data of PHANTOM2. The results show that the reconstruction quality of the MR-FBP method when reconstructing truncated data is similar to the quality achieved when reconstructing untruncated data.

Comparison to gaussian filtered FBP

A popular improvement to the standard FBP method is to apply a gaussian filter to the image after reconstruction. Since different filter widths can be used, there is a certain trade-off between the amount of artifacts and the sharpness in the resulting image. By using a wider filter, the artifacts of the FBP reconstructions are suppressed more, but the resulting image will be more blurred as well. In Fig. 2.19, a comparison

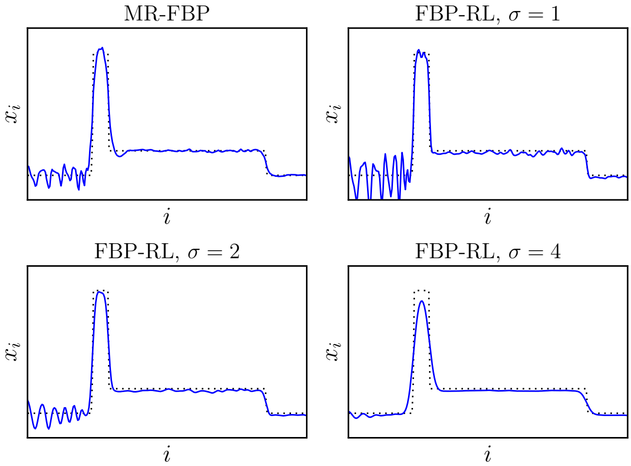


Figure 2.19: Partial line profiles of reconstructions of PHANTOM1 projection data of 64 projections, produced by MR-FBP and standard FBP with gaussian filtering with different window widths (σ values). The partial line profile of the ground truth, the PHANTOM1 image, is shown by a dotted line.

between the MR-FBP method and the FBP method with gaussian filtering is made by examining a partial line profile of reconstructions of PHANTOM1. The results show that the MR-FBP reconstruction matches the ground truth better, even with a similar level of artifact reduction. The mean absolute error of the different methods for the entire reconstruction is 0.023, 0.036, 0.029, and 0.027 for MR-FBP and FBP with $\sigma = 1$, $\sigma = 2$, and $\sigma = 4$, respectively. Figure 2.19 also shows that the reconstruction quality of the gaussian filtered FBP reconstructions depends highly on the chosen window width. For a given dataset, it can be hard to determine the correct width to use, since it depends on the number of projections and the amount of noise present in the projection data. Furthermore, the parameters are often determined by visual inspection, which is subjective by nature. The MR-FBP method, on the other hand, determines the filter by minimizing a well-defined metric, making it both objective and reproducible, and automatically adjusts the filter to the projection data and geometry, as shown in Fig. 2.16.

2.8 Conclusion

In this chapter, we introduced a novel reconstruction method for 2D parallel-beam tomographic reconstruction problems, the minimum residual filtered backprojection method (MR-FBP). A reconstruction calculated by MR-FBP is a standard FBP reconstruction with a custom filter. For each reconstruction, a new filter is calculated by minimizing the squared difference of the projections of reconstructed image with the measured projections. Since FBP is a linear operation on the projection data, the optimal filter can be found by solving a linear system in a least squares sense, similar to algebraic reconstruction techniques. In this chapter, we showed that the linear system can be constructed efficiently column by column. The result is an efficient method that automatically determines a data-dependent filter based on an objective quality criterion, eliminating the need for subjective manual filter selection. Furthermore, the method can be extended to exploit certain forms of prior knowledge to improve reconstruction quality.

Results for both phantom data and experimental data show that MR-FBP is able to produce more accurate reconstructions than FBP with standard filters when presented with data from few projections. The reconstructed images obtained from MR-FBP are visually similar to SIRT, an algebraic method, but can be calculated significantly faster. Similar results are found when reconstructing from projection data with various amounts of Poisson noise applied. Additional experiments show that exponential binning does not decrease reconstruction quality, while reducing computation time significantly.

We showed that MR-FBP can be extended to incorporate certain forms of regularization based on features of the image to improve reconstruction quality. One example, where a term minimizing the gradient of the reconstructed image is added to MR-FBP, was examined in detail. Results for the resulting method, called MR-FBP$_{\mathrm{GM}}$, show that by exploiting prior knowledge we indeed obtain more accurate reconstructions from noisy projection data compared to methods that do not exploit prior knowledge. Furthermore, because MR-FBP$_{\mathrm{GM}}$ is an extension of the MR-FBP method, it is able to calculate reconstructions in less time than SIRT.

The results from this chapter show that by exponential binning we are able to reduce the computation time of the MR-FBP method without reducing its reconstruction quality. Other bases for reducing the number of unknowns in the linear system can be used however, which might enable us to reduce computation time even further, or improve reconstruction quality. For example, it might be possible to include some prior knowledge about the scanned object in the choice of basis. Whether better bases can be found, and whether they are useful in practice, is subject to further research.

The current study focused on two-dimensional parallel beam tomography. An approach similar to the one used in this chapter can, however, also be applied to other tomographic settings for which linear filter-based reconstruction methods exist. For example, in three-dimensional cone-beam settings, a minimum residual Feldkamp-David-Kress (FDK) method could be formulated, based on the standard FDK reconstruction method [FDK84]. The reconstruction quality of such methods is subject to further research.

In this chapter, we showed that the linear system of MR-FBP can be calculated column-by-column by creating several FBP reconstructions with specific filters. Therefore, MR-FBP can be implemented relatively easy using existing FBP implementations. Furthermore, if an optimized implementation of FBP is available, MR-FBP can use this implementation, resulting in an optimized version of MR-FBP as well. Therefore, MR-FBP can be used relatively easily to improve reconstruction quality in practical applications.

3

Approximating SIRT by filtered backprojection

3.1 Introduction

In computed tomography, two common approaches to reconstruct objects from their projections are *analytical* methods and *algebraic* methods. Analytical reconstruction is based on inverting a continuous model of the problem, and discretizing the result. The popular filtered backprojection method for parallel-beam projections is a result of this approach [KS01], as well as the FDK method for cone-beam projection data [FDK84]. Analytical methods assume that projection data is available for *all* angles, and that the data is free of noise. In many practical applications, it is either impossible or undesirable to acquire a sufficient number of noise-free projections to accurately reconstruct the scanned object with an analytical method. In such cases, algebraic methods are often able to yield more accurate reconstructions.

Algebraic methods are based on a discretized model of the problem, resulting in a linear system of equations. A reconstruction is then computed by solving this linear system using an iterative method. Since algebraic methods are based on a model of the data that is available instead of assuming perfect data, algebraic reconstructions are often of higher quality than analytical reconstructions when presented with a limited

---

This chapter is based on:

D. M. Pelt and K. J. Batenburg. “Accurately approximating algebraic tomographic reconstruction by filtered backprojection”. In: *Proceedings of the 2015 International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Ed. by M. King, S. Glick, and K. Mueller. 2015, pp. 158–161.

number of projections. Furthermore, the effect of noise on the reconstruction can be minimized by using certain forms of regularization in the iterative method.

Despite the advantages of algebraic methods for imperfect data, analytical methods remain very popular in practice. In [PSV09], several reasons for the popularity of analytical methods are discussed, one of which is the gap that exists between a mathematical definition of an algebraic method and its application in actual real-world problems. When implementing an algebraic method in real-world applications, many difficulties can arise, for example with computational requirements, which are typically much higher compared to analytical methods. In this chapter, we aim to bridge the gap by introducing a method that approximates the algebraic SIRT method by computing a special filter for the filtered backprojection (FBP) method. The resulting method can achieve a reconstruction quality similar to algebraic methods using existing, computationally efficient, FBP implementations.

Recently, a number of reconstruction methods have been proposed that aim to improve FBP by changing its convolution filter. In [Zen12], a window function for the standard ramp filter is derived that approximates an algebraic method. During the derivation, however, it is assumed that projection data is available for enough angles such that a certain approximation can be made, which may not be the case in practice. A different approach is taken in Chapter 2 of this thesis, where a data-dependent filter is computed that minimizes the projection error of the resulting FBP reconstruction. Since a different filter has to be computed for each scanned object, the computational requirements of this method are higher than for standard FBP. Finally, in [BP12], a way of computing angle-dependent filters that approximate algebraic methods is proposed. The method for computing the filters is very computationally demanding, however, which severely limits its applicability in practice. In this chapter, we propose a method to compute filters that are similar to those in [BP12], but can be computed much faster, using an approach that is, in part, similar to [Zen12].

3.2 Method

In this section, we propose a method for computing filters for the parallel-beam FBP method that approximate the algebraic SIRT method [KS01], which is a method from the class of Landweber iteration methods [Lan51]. We assume that projection data is acquired for N_θ angles, with N_d detector elements per projection. In this case, we can write the acquired projection data as a vector $\boldsymbol{p} \in \mathbb{R}^{N_\theta N_d}$. Similarly, we can write the reconstructed image, which is defined on a $N \times N$ pixel grid, as a vector $\boldsymbol{x} \in \mathbb{R}^{N^2}$. An element w_{ij} of the *projection matrix* $\boldsymbol{W}$ gives the contribution of pixel j to detector element i. Using these definitions, we can write the FBP method as:

$$FBP(\boldsymbol{p},\boldsymbol{h}) = \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \tag{3.1}$$

Here, $\boldsymbol{C}_{\boldsymbol{h}}$ is a convolution of each detector with filter $\boldsymbol{h}$, which can be angle-dependent. Note that in parallel-beam tomography, an FBP reconstruction can also be calculated by first backprojecting the projections, and filtering the result:

$$FBP(\boldsymbol{p},\boldsymbol{h}') = \boldsymbol{H}_{\boldsymbol{h}'} \boldsymbol{W}^T \boldsymbol{p} \tag{3.2}$$

Here, $\boldsymbol{H}_{\boldsymbol{h}'}$ is a two-dimensional convolution of an image with filter $\boldsymbol{h}'$, and $\boldsymbol{h} = \boldsymbol{W}\boldsymbol{h}'$.

The standard definition of the SIRT method is the following iterative equation:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha \boldsymbol{W}^T \left(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}^k\right) \tag{3.3}$$

Here, $\boldsymbol{x}^k$ is the reconstructed image at iteration k, and α is a parameter that influences the stability and convergence of the method, for which the standard value of $\alpha = (N_\theta N_d)^{-1}$ is used throughout this chapter. By regrouping terms, we can write this equation in a matrix form:

$$\boldsymbol{x}^{k+1} = \left(\boldsymbol{I} - \alpha \boldsymbol{W}^T \boldsymbol{W}\right)\boldsymbol{x}^k + \alpha \boldsymbol{W}^T \boldsymbol{p} \tag{3.4}$$

Note that this is a recurrence relation of the form $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{b}$, which has as solution for the reconstruction at iteration n:

$$\boldsymbol{x}^n = \boldsymbol{A}^n \boldsymbol{x}^0 + \alpha \left[\sum_{i=0}^{n-1} \boldsymbol{A}^i\right] \boldsymbol{W}^T \boldsymbol{p} \tag{3.5}$$

Here, $\boldsymbol{A} = \boldsymbol{I} - \alpha \boldsymbol{W}^T \boldsymbol{W}$. In many cases, a zero image is used as the initial image $\boldsymbol{x}^0$, in which case Eq. (3.5) becomes:

$$\boldsymbol{x}^n = \alpha \left[\sum_{i=0}^{n-1} \boldsymbol{A}^i\right] \boldsymbol{W}^T \boldsymbol{p} \tag{3.6}$$

The similarities between Eq. (3.6) and Eq. (3.2) suggest that we can approximate the SIRT equation (Eq. (3.6)) by approximating $\sum_{i=0}^{n-1} \boldsymbol{A}^i$ by a two-dimensional convolution operation with filter $\boldsymbol{q}_n$:

$$\boldsymbol{x}^n \approx \alpha \boldsymbol{H}_{\boldsymbol{q}_n} \boldsymbol{W}^T \boldsymbol{p} \tag{3.7}$$

To find a good approximating filter, we can take the impulse response of $\sum_{i=0}^{n-1} \boldsymbol{A}^i$:

$$\boldsymbol{q}_n = \sum_{i=0}^{n-1} \boldsymbol{A}^i [0, \dots, 0, 1, 0, \dots, 0]^T \tag{3.8}$$

In other words, we start with an image with the central pixel set to one and the other pixels set to zero, and iteratively apply $\boldsymbol{A}$ to the image n times, summing all images along the way. In parallel-beam tomography, we can write Eq. (3.7) in the standard FBP form by forward projecting $\boldsymbol{q}_n$:

$$\boldsymbol{x}^n \approx \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{u}_n} \boldsymbol{p} = FBP(\boldsymbol{p}, \boldsymbol{u}_n) \tag{3.9}$$

$$\boldsymbol{u}_n = \alpha \boldsymbol{W} \boldsymbol{q}_n \tag{3.10}$$

We conclude that we can approximate the algebraic SIRT method by the FBP method with filter $\boldsymbol{u}_n$. The filter is computed by first computing $\boldsymbol{q}_n$ by applying $\boldsymbol{A}$ to a certain image n times, summing the results. The resulting image is forward projected to obtain $\boldsymbol{u}_n$. Note that a single computed filter can be used to reconstruct many different objects,

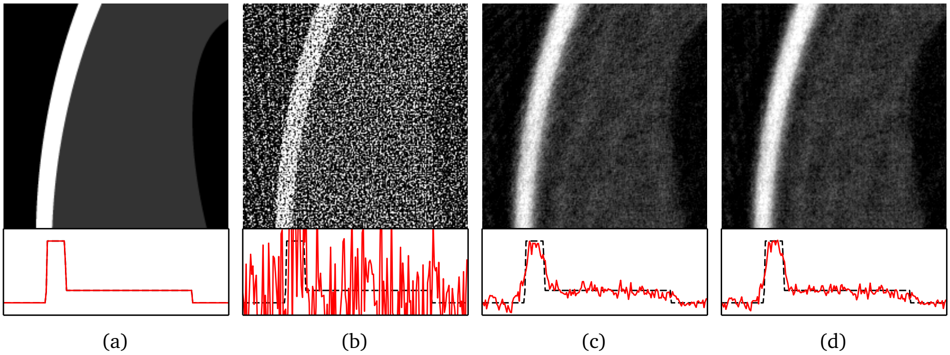


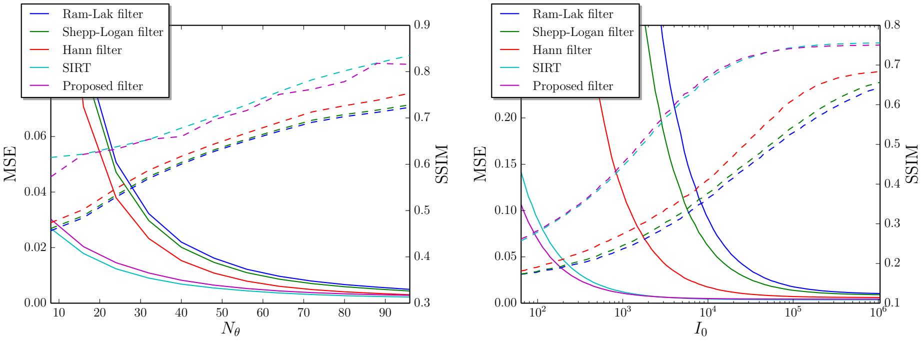
Figure 3.1: Cropped reconstructions of the Shepp-Logan head phantom (a) obtained by different reconstruction methods: FBP with the standard Ram-Lak filter (b), SIRT (c), and FBP with the proposed filter (d). The reconstructions are computed using simulated projection data of 64 angles, with 1024 samples per projection (rebinned from 4096 samples) and Poisson noise applied. A line profile of the central line of each cropped reconstruction is shown under each image, with the phantom shown by a dashed line.

as long as they are scanned with the same projection geometry. For computing the filter, $2n+1$ projection operations are needed, which is similar in computation time to a single run of the SIRT method. To compare, $\mathcal{O}(N_\theta N_d)$ runs of an algebraic method are needed to compute a similar filter in [BP12].

By approximating $\boldsymbol{A}$ by a convolution operation, we assume that the $\boldsymbol{W}^T\boldsymbol{W}$ operation is approximately shift-invariant. Whether this assumption is correct can depend on the actual implementation of the projection operator $\boldsymbol{W}$. If a ray is defined as a strip with the same width as a detector pixel, and the weight of w_{ij} is given by the area of the intersection of the pixel j and the ray i, $\boldsymbol{W}^T\boldsymbol{W}$ can be well approximated by a shift-invariant operation. If a ray is defined as a line of zero thickness, the approximation is not as accurate. In this case, however, the approximation can be improved by using supersampling, where multiple lines of zero thickness are cast through the volume per detector pixel. Note that supersampling is only needed during computation of the filter, and not during reconstruction using Eq. (3.9). In the rest of this chapter, we cast eight lines per detector pixel during computation of the filters.

3.3 Experiments

We implemented the proposed filter calculation method in Python 3.3.2 using the ASTRA toolbox [PBS11], which includes projection operations that use graphic processing units (GPUs) to improve performance. All reconstructions presented in this chapter are calculated by the ASTRA toolbox as well. To investigate the reconstruction quality of the proposed method compared to other reconstruction methods, we use reconstructions of the Shepp-Logan head phantom. For each experiment the phantom is generated on a 4096 × 4096 pixel grid, for which projections are simulated with 4096 detector elements per projection. The resulting projection data is rebinned to 1024 detector



(a) (b)

Figure 3.2: MSE (solid) and SSIM (dashed) of reconstructions of the Shepp-Logan head phantom computed by various methods. In (a), the metrics are shown as a function of the number of projections N_θ, for noise-free projection data. In (b), the metrics are shown for reconstructions of 64 projections, as a function of the amount of applied Poisson noise (I_0). Higher values of I_0 correspond to lower amounts of applied noise.

elements per projection, which are used to reconstruct on a 1024×1024 pixel grid. The reconstructions are compared to the phantom, rebinned to 1024×1024 pixels. We give the mean squared error (MSE) and the structural similarity index (SSIM) [Wan+04] of each reconstruction compared to the rebinned phantom. For the error measures, we use all pixels that are within a disc with a radius of 512 pixels, centered in the pixel grid. For all experiments, we compute a filter that approximates 200 iterations of the SIRT method, unless stated otherwise. Each SIRT reconstruction is computed using 200 iterations as well.

In Fig. 3.1, cropped reconstructions of the Shepp-Logan phantom are shown for FBP with the standard Ram-Lak filter, SIRT, and FBP with the proposed filter, computed using data of 64 projections with a moderate amount of applied Poisson noise. The results for FBP (Fig. 3.1b) show that the noise present in the FBP reconstruction can be prohibitive for further analysis. The reconstructions of SIRT (Fig. 3.1c) and FBP with the proposed filter (Fig. 3.1d) are, at least visually, very similar.

To further investigate the reconstruction quality of the proposed method, we generated projection data for different numbers of projections N_θ, and compared the MSE and SSIM of reconstructions of FBP with different standard filters, SIRT, and the proposed filter method. The results are shown in Fig. 3.2a. We also generated data of 64 projections with various amounts of applied Poisson noise, for which the results are given in Fig. 3.2b. Here, I_0 indicates the amount of applied Poisson noise, with higher values corresponding to lower amounts of applied noise. In both Fig. 3.2a and Fig. 3.2b, reconstructions using the proposed method have a significantly lower MSE and higher SSIM compared to FBP reconstructions with standard filters. Compared to SIRT, the proposed method has a similar MSE and SSIM. Note, however, that for 64 projections a SIRT reconstruction took 1.40 seconds to compute, while a reconstruction of the proposed method was computed in 9.67 milliseconds, which is roughly 144×

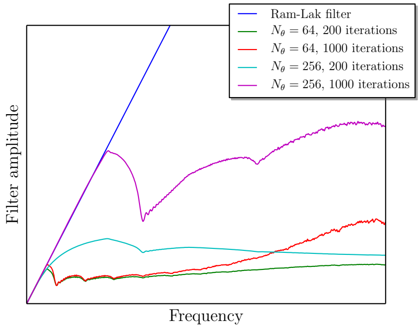


Figure 3.3: Comparison of the standard Ram-Lak filter and the computed filters for $N_d = 1024$ and various numbers of projections N_θ and numbers of iterations, averaged over all angles, shown in Fourier space.

faster.

Computed filters, averaged over all angles, are shown in Fourier space in Fig. 3.3, along with the standard Ram-Lak filter. The figure shows that the computed filters are identical to the Ram-Lak filter up to a certain frequency, which depends on the number of projections and iterations. Taking more angles and/or iterations results in a filter that is closer to the Ram-Lak filter. The figure also shows that by taking more iterations, the higher frequencies of the filters are amplified. A similar effect can also be observed in the SIRT method, where taking more iterations results in stronger high frequencies in the reconstructed image.

3.4 Conclusion

In this chapter, we presented a novel method to compute filters for the filtered backprojection method that approximate the algebraic SIRT method. The method is based on rewriting SIRT into a matrix form, and approximating the combined backprojection and forward projection operation ($\boldsymbol{W}^T\boldsymbol{W}$) by a 2D convolution operation. An approximating filter can be found by applying the combined projection operation repeatedly to a specific image. The result is an angle-dependent filter that can be used in the FBP method to produce reconstructions that are similar to those produced by SIRT. Computation of the filter is significantly faster than in similar approaches of previous work [BP12], enabling its use in large-scale real-world tomographic problems.

Several experiments on a phantom image show that the proposed method produces reconstructions of similar quality to the SIRT method, both in the case of a low number of projections and with noise present in the data. Compared to FBP with standard filters, the proposed method produces reconstructions with significantly lower MSE and higher SSIM. The computation time of reconstructing with the proposed method is identical to the FBP method, which is significantly lower than SIRT. These results show that by computing geometry-dependent convolution filters, it is possible to accurately approximate the SIRT method by filtered backprojection.

4

Local approximation of advanced regularized iterative methods

4.1 Introduction

The goal of tomography is to reconstruct an object given its projections for different angles. Using tomography, it is possible to nondestructively examine the interior of objects, which makes it useful for many applications. Examples of tomography in practice include computed tomography in medicine and electron tomography in materials science. Because of its practical usefulness, many algorithms have been developed to perform tomographic reconstruction. An overview of past research on tomography can be found in [KS01; Nat01; Buz08]. Two types of reconstruction methods are commonly used: *analytical* methods, which discretize a continuous inversion formula of the problem, and *algebraic* methods, in which a linear system that represents the problem is solved.

In many applications of tomography, it is impossible to acquire a large number of noise-free projections. For example, when scanning live animals, there is a limit on the total dose deposited on the animal during the experiment [Lov+13]. In electron

---

This chapter is based on:

D. M. Pelt and K. J. Batenburg. "A method for locally approximating advanced regularized iterative tomographic reconstruction methods". *IEEE Transactions on Image Processing* (Submitted for publication).

tomography, the scanned sample is damaged by the electron beam, which leads to a limit on the number of projections that can be acquired [MDG95]. In these cases, standard reconstruction methods often fail to produce reconstructions with adequate quality for further analysis [Lov+13]. For analytical methods, the reason for this is that the continuous inversion formulas on which they are based assume that noise-free projections are available for *all* angles. In algebraic methods, the linear system that is solved is typically both underdetermined and ill-conditioned, which can make it difficult to find accurate reconstructions when the available projection data is limited and/or noisy.

Recently developed advanced reconstruction methods aim to improve reconstruction quality by exploiting prior knowledge about the scanned object or scanning system. Often, these methods add additional terms to the objective function that is minimized in standard algebraic methods. Methods of this type will be called *regularized iterative methods* in this chapter. For example, if it is known beforehand that the physical quantity that is reconstructed cannot be negative, a nonnegativity constraint can be added to the objective function to improve the reconstruction quality. If it is known that the scanned object has a sparse boundary, total variation minimization can be applied by adding a term that minimizes the gradient of the reconstructed image [SP08]. If the added prior knowledge is appropriate for the acquired data, regularized iterative methods can be extremely successful in reconstructing objects from (highly) limited data [BS11; Kos+13].

One of the main disadvantages of regularized iterative methods is their computational cost, which is typically very high. A high computational cost of a reconstruction method can be prohibitive for its application in practice. For example, in ultrafast tomographic experiments at synchrotrons, the computation time of the reconstruction method has to match the high speed of the acquisition of projection data [Mok+13]. An additional problem is that regularized iterative methods often have a number of tunable parameters that influence the reconstruction quality greatly. In many cases, values for these parameters are chosen by trial-and-error, which can be very time-consuming for methods with a high computational cost. These problems are especially important in cases where a large object is scanned, but the features of interest are only located in a small region of the object. Since regularized iterative methods, and the algebraic methods they are based on, minimize a *global* objective function, they typically need to compute the entire volume during reconstruction, which may not fit into the available memory of the graphic processing units used to perform the reconstruction [XM05].

Analytical methods, on the other hand, can be evaluated *locally*: if one is only interested in a small subvolume of the reconstruction, only that subvolume has to be reconstructed. When reconstructing large volumes, analytical methods can divide the reconstruction volume into subvolumes that do fit into the available memory, and reconstruct each subvolume separately, resulting in an efficient method to compute the full reconstruction volume. This property is one of the reasons that in many applications of tomography, standard analytical methods are still the most popular reconstruction methods instead of regularized iterative methods [PSV09].

In this chapter, we present a novel method for approximating a computationally expensive regularized iterative method in a (small) subvolume of the full reconstruction volume. The proposed method only performs computations in the chosen subvolume, ensuring low computational and memory requirements. If one is only interested in part of the scanned object, the new method can significantly reduce the time needed to reconstruct that part compared to existing regularized iterative methods. If one wants to reconstruct the entire object, the proposed method also allows for significant reduction of computation time by enabling parallel computation of different subvolumes, and it enables regularized iterative reconstruction of large datasets that do not fit completely into the available memory. In addition, the method can be used to quickly estimate parameters of a slow regularized iterative method by estimating them in a small subvolume.

The proposed method is based on approximating standard algebraic methods by a modified analytical method. In recent years, several methods have been proposed that achieve this by modifying the filter that is typically used in analytical methods. In one study, an angle-independent filter is calculated based on analytic analysis of the algebraic SIRT method [Zen12]. An extension of the method for noisy projection data is given in [ZZ13]. In Chapter 2 of this thesis, a method of calculating a data-dependent filter is given. Finally, an angle-dependent and geometry-dependent filter is calculated by repeated application of the SIRT method in [BP12]. A faster method of calculating similar filters for the algebraic SIRT method is proposed in Chapter 3 of this thesis. None of these methods, however, allow for inclusion of popular advanced prior knowledge terms, such as total variation minimization, which can limit their usefulness in practice. In this chapter, we first show the application of the filter of Chapter 3 to locally approximate the algebraic SIRT method. Then, we extend the method to allow for local approximation of a regularized iterative method as well. Finally, we demonstrate that the proposed method is able to produce local reconstructions that are very similar to reconstructions of global regularized iterative methods for various types of exploited prior knowledge.

This chapter is structured as follows. In Section 4.2, we introduce the notations we use throughout the chapter, and formally define the tomographic reconstruction problem and the standard analytical and algebraic approaches. The main contribution of this chapter is given in Section 4.3, where we first apply the method proposed in Chapter 3 to approximate SIRT locally. We then extend this approximation by including prior knowledge in the reconstruction of a subvolume, and give some details on how to implement the resulting method in practice. The experiments we performed to study the new method are explained in Section 4.4, and the results of those experiments are shown in Section 4.5. We conclude in Section 4.6 with a brief summary of the chapter and some final remarks.

4.2 Notation and concepts

In this section, the mathematical notation that we use throughout the chapter is introduced, and a formal definition of the tomographic reconstruction problem is given. The standard analytical and algebraic approaches to the problem are explained, and their mathematical definitions are given. Finally, we explain how prior knowledge can

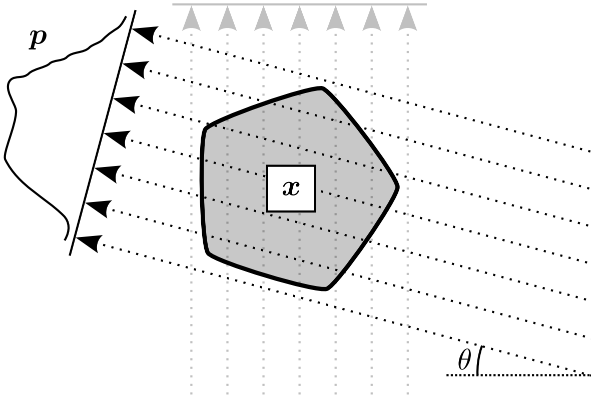


Figure 4.1: The two-dimensional parallel-beam geometry used in this chapter. Parallel lines, rotated by angle θ, pass through the object f. A line $l_{\theta,t}$ has the characteristic equation $t = x\cos\theta + y\sin\theta$, and a projection $P_\theta(t)$ of f is given by the line integral of f over the line $l_{\theta,t}$.

be exploited in algebraic methods by extending their objective functions, resulting in regularized iterative methods.

Notation and problem definition

We focus on two-dimensional parallel-beam tomographic reconstruction problems with a single rotation axis. Note that in many cases it is possible to convert other tomographic geometries, such as cone-beam or spiral tomography, to a parallel-beam geometry by rebinning [GKP00; KSK00]. Parallel-beam projection data are acquired by rotating an array of detectors around the object (or, equivalently, rotating the object), with the detectors of the array located on a straight line. This acquisition scheme is shown graphically in Fig. 4.1. If the number of detectors in the array is denoted by N_d, and the number of rotation angles for which data are acquired is denoted by N_θ, we can write the measured line integrals as a vector $\boldsymbol{p}$ with $N_d N_\theta$ elements, one for each combination of detector element and rotation angle. The reconstructed image is represented as a vector $\boldsymbol{x}$ with N^2 elements, one for each pixel of the $N \times N$ pixel grid on which the reconstruction is calculated. The main problem in tomographic reconstruction is to find the unknown image $\boldsymbol{x}$, given the acquired projection data $\boldsymbol{p}$.

The *forward projection* operator $\mathcal{W} : \mathbb{R}^{N^2} \rightarrow \mathbb{R}^{N_d N_\theta}$ is the operator that, for a given projection geometry, corresponds to the discretized line integrals of an object represented on a $N \times N$ pixel grid. Using the above notation, we can write this operator as a $N_d N_\theta \times N^2$ matrix $\boldsymbol{W}$, with element w_{ij} giving the contribution of pixel j to detector i. The transpose of this operator, $\boldsymbol{W}^T$, is called the *backprojection* operator. Typically, a forward projection of an image $\boldsymbol{x}$ is calculated on-the-fly by calculating its line integrals directly [PBS11]. Similarly, multiplying $\boldsymbol{p}$ by $\boldsymbol{W}^T$ is done implicitly by backprojecting $\boldsymbol{p}$ on-the-fly. The advantage of this approach is that the matrix $\boldsymbol{W}$, which can be very large, never has to be stored in memory. Furthermore, forward projections and backprojections can be computed very efficiently on graphic processor units (GPUs) [XM05; MXN07].

Our novel approach aims to reconstruct only a *local* part $\mathscr{L}$ of the entire reconstruction grid. Here, $\mathscr{L}$ is a subset of all N^2 pixels of the entire reconstruction grid, usually ordered in a $N_{\mathscr{L}} \times N_{\mathscr{L}}$ grid as well. Let $\boldsymbol{M}_{\mathscr{L}}$ be a diagonal matrix with a value 1 on the diagonal of row i if pixel i is inside $\mathscr{L}$, and 0 everywhere else. In other words, $\boldsymbol{M}_{\mathscr{L}}$ keeps all pixels of an image that are inside $\mathscr{L}$, and zeros all other pixels. Similarly, we define a matrix $\boldsymbol{M}_{\mathscr{F}}$ that zeros all pixels inside $\mathscr{L}$, and keeps all other pixels. Using these, we can define local operators $\boldsymbol{W}_{\mathscr{L}}$ and $\boldsymbol{W}_{\mathscr{L}}^T$, and outer operators $\boldsymbol{W}_{\mathscr{F}}$ and $\boldsymbol{W}_{\mathscr{F}}^T$:

$$\begin{aligned} \boldsymbol{W}_{\mathscr{L}} &= \boldsymbol{W}\boldsymbol{M}_{\mathscr{L}} \\ \boldsymbol{W}_{\mathscr{L}}^T &= \boldsymbol{M}_{\mathscr{L}}\boldsymbol{W}^T \\ \boldsymbol{W}_{\mathscr{F}} &= \boldsymbol{W}\boldsymbol{M}_{\mathscr{F}} \\ \boldsymbol{W}_{\mathscr{F}}^T &= \boldsymbol{M}_{\mathscr{F}}\boldsymbol{W}^T \end{aligned} \tag{4.1}$$

Since $\boldsymbol{M}_{\mathscr{L}} + \boldsymbol{M}_{\mathscr{F}} = \boldsymbol{I}$ by construction, we have that the sum of $\boldsymbol{W}_{\mathscr{L}}$ and $\boldsymbol{W}_{\mathscr{F}}$ is equal to $\boldsymbol{W}$:

$$\boldsymbol{W} = \boldsymbol{W}_{\mathscr{L}} + \boldsymbol{W}_{\mathscr{F}} \tag{4.2}$$

Note that local forward projections and backprojections can be computed significantly faster than full forward projections and backprojections, since many rows and columns of $\boldsymbol{W}_{\mathscr{L}}$ and $\boldsymbol{W}_{\mathscr{L}}^T$ are zero.

Common reconstruction methods

Using the above definitions, we can write one of the most popular reconstruction methods, the analytical filtered backprojection (FBP) method, as:

$$\mathit{FBP}(\boldsymbol{p}, \boldsymbol{h}) = \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \tag{4.3}$$

Here, $\boldsymbol{C}_{\boldsymbol{h}}$ is a convolution operator that convolves each 1D array of detector values, taken at a single rotation angle, with the 1D filter $\boldsymbol{h}$ [KS01]. Note that this 1D filter can be different for each rotation angle. Several fixed angle-independent filters are commonly used in practice, such as the Ram-Lak (ramp), Shepp-Logan, and Hann filters [Far+97]. One reason for the popularity of FBP is its computational efficiency: the filtering step can be performed very efficiently in Fourier space, and only one backprojection has to be computed during reconstruction. Another advantage of the filtered backprojection method compared to other methods is that we can calculate its values inside the local part $\mathscr{L}$ by simply exchanging $\boldsymbol{W}^T$ by $\boldsymbol{W}_{\mathscr{L}}^T$ in Eq. (4.3):

$$\mathit{FBP}_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{h}) = \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{C}_{\boldsymbol{h}} \boldsymbol{p} \tag{4.4}$$

A different approach to solving the reconstruction problem is the algebraic approach. Here, we form a linear system $\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}$, and solve for $\boldsymbol{x}$. Most algebraic methods find a solution $\boldsymbol{x}_{alg}$ by minimizing the difference, in some vector norm, between the forward projection of the solution and the measured projection data. This difference is called the *projection error*. In the case of the ℓ_2-norm, we can write this as:

$$\boldsymbol{x}_{alg} = \operatorname*{argmin}_{\boldsymbol{x}} \|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 \tag{4.5}$$

Since the matrix $\boldsymbol{W}$ is often very large, Eq. (4.5) is usually not solved directly. Instead, an iterative optimization method is typically used to iteratively decrease the projection error. Implicit regularization of the solution can be included by stopping the iteration process early, which is needed because $\boldsymbol{W}$ is usually ill-conditioned and noise is often present in $\boldsymbol{p}$.

Different iterative optimization methods can be used to minimize the projection error, leading to different algebraic methods. The CGLS method, for example, is based on a conjugate gradient method [Bjö96]. Another popular algebraic method is the simultaneous iterative reconstruction technique (SIRT) [KS01]. The SIRT method belongs to the class of Landweber iteration methods [Lan51], and uses a specific Krylov subspace method to minimize the projection error iteratively. A single iteration of the SIRT method can be viewed as a gradient-descent step on the projection error, and can be written as:

$$\boldsymbol{x}_s^{k+1} = S(\boldsymbol{x}_s^k) = \boldsymbol{x}_s^k + \alpha \boldsymbol{W}^T \left(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}_s^k\right) \tag{4.6}$$

Note that in algebraic methods, we are not able to simply exchange $\boldsymbol{W}$ by $\boldsymbol{W}_{\mathscr{L}}$ to find the reconstruction inside $\mathscr{L}$, since then we would be solving the linear system $\boldsymbol{W}_{\mathscr{L}}\boldsymbol{x} = \boldsymbol{p}$, which will have a completely different solution than $\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}$ if the scanned object is nonzero outside $\mathscr{L}$.

Regularized iterative methods

A common way of including prior knowledge in algebraic methods is to add additional constraints to the objective function of Eq. (4.5). In this chapter, we distinguish two types of constraints that are commonly used: *domain constraints*, which restrict the domain of possible solutions, and *penalty constraints*, which penalize undesired solutions in the objective function. The resulting regularized iterative methods can be written as:

$$\boldsymbol{x}_{reg} = \operatorname*{argmin}_{\boldsymbol{x}\in D} \left[\|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2 + \lambda g(\boldsymbol{x}) \right] \tag{4.7}$$

Here, D is a restricted domain for the possible solutions $\boldsymbol{x}$, and $g : \mathbb{R}^{N^2} \rightarrow \mathbb{R}$ is a penalty function that penalizes solutions that do not fit with the assumed prior knowledge. The λ term controls how strongly the penalty function is weighted compared to the projection error term. The domain D is used to specify domain constraints, for example when adding a nonnegativity constraint on the values of $\boldsymbol{x}$ by using $D = \{\boldsymbol{x} \in \mathbb{R}^{N^2}; x_i \geq 0,\ i = 1, \ldots, N^2\}$. The cost function $g(\boldsymbol{x})$ is used to specify penalty constraints. For example, if we assume that the scanned object is sparse in some wavelet basis, we can set $g(\boldsymbol{x}) = \|\boldsymbol{B}\boldsymbol{x}\|_1$, where $\boldsymbol{B}$ is the wavelet decomposition operator. Similarly, if we assume that the gradient of the scanned object is sparse, we set $g(\boldsymbol{x}) = \|\nabla \boldsymbol{x}\|_1$ to perform total variation minimization, where ∇ is a discrete

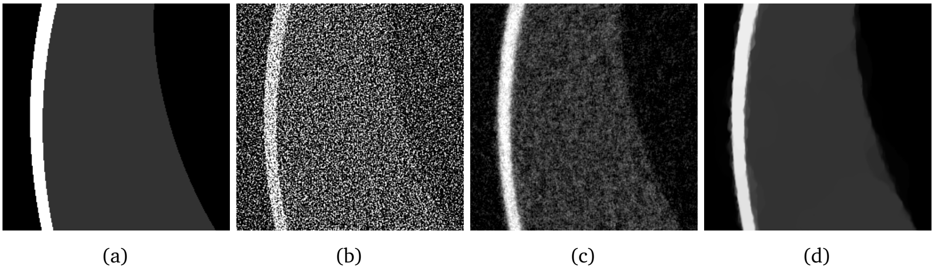


Figure 4.2: Zoomed-in reconstructions of the Shepp-Logan head phantom (a), showing the resulting images of three different reconstruction methods: (b) FBP, (c) SIRT, and (d) total variation minimization. The images were reconstructed on a 1024 × 1024 pixel grid, using projection data acquired with $N_d = 1024$ detectors and $N_\theta = 256$ projection angles, equally distributed in the interval $[0, \pi]$, and additional Poisson noise applied.

gradient operator. Several algorithms exist that are able to find solutions to Eq. (4.7), such as the popular fast iterative shrinkage-thresholding algorithm (FISTA) [BT09a], Chambolle-Pock algorithms [CP11], and adaptive steepest descent projection onto convex sets algorithm (ASD-POCS) [SP08]. A comparison of reconstructions obtained using FBP, SIRT, and total variation minimization from noisy projection data is shown in Fig. 4.2.

Many regularized iterative methods use a scheme that alternates between gradient-descent steps on the projection error $\|\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}\|_2^2$, steps that minimize the penalty function $g(\boldsymbol{x})$, and steps that enforce the domain constraints D. Since a single iteration of the SIRT method is identical to a single gradient-descent step on the projection error, these regularized iterative methods can be viewed as a combination of SIRT iterations and some additional steps incorporating the prior knowledge. As an example, one can include box constraints on the values of the reconstruction pixels of the form $l \le x_i \le r$, which is a domain constraint with $D = \{\boldsymbol{x} \in \mathbb{R}^{N^2}; l \le x_i \le r,\ i = 1, \ldots, N^2\}$ by using the following iterations for pixel i of the reconstruction:

$$\boldsymbol{x}_i^{k+1} = \begin{cases} l & : \text{ if } S(\boldsymbol{x}^k)_i < l \\ r & : \text{ if } S(\boldsymbol{x}^k)_i > r \\ S(\boldsymbol{x}^k)_i & : \text{ otherwise} \end{cases} \tag{4.8}$$

An example of using a penalty constraint is the ISTA method [DDD04] for ℓ_1-norm minimization of a representation of the reconstructed image in a wavelet basis. In this case, a single iteration of the method can be written as:

$$\boldsymbol{x}^{k+1} = \boldsymbol{B}^{-1}\mathscr{P}_\lambda(\boldsymbol{B}S(\boldsymbol{x}^k)) \tag{4.9}$$

where $\boldsymbol{B}$ is the wavelet decomposition operator, and $\mathscr{P}_\lambda$ the soft thresholding operator with threshold λ:

$$\mathscr{P}_\lambda(\boldsymbol{y})_i = \begin{cases} \operatorname{sgn}(\boldsymbol{y}_i)(|\boldsymbol{y}_i| - \lambda) & : \text{ if } \boldsymbol{y}_i > \lambda \\ 0 & : \text{ otherwise} \end{cases} \tag{4.10}$$

In this chapter, we propose a method to locally approximate regularized iterative reconstruction methods that are a combination of SIRT iterations and additional steps that incorporate the prior knowledge.

4.3 Method

In this section, we introduce the major contribution of this chapter: a local approximation method for regularized iterative reconstruction methods. We first explain the method introduced in Chapter 3 to approximate the algebraic SIRT method by FBP with a specific geometry-dependent filter, and show how this approach can be used to approximate SIRT locally as well. Afterwards, we extend the approximation to include prior knowledge, improving the reconstruction quality. Finally, we give details on how we implemented the resulting method for the experiments of Section 4.4.

Local approximation of SIRT

Recall that a single iteration of the SIRT method can be written as:

$$\boldsymbol{x}_s^{k+1} = S(\boldsymbol{x}_s^k) = \boldsymbol{x}_s^k + \alpha \boldsymbol{W}^T \left(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}_s^k\right) \tag{4.6}$$

Here, $\alpha \in \mathbb{R}$ is a parameter that influences the stability and rate of convergence of the method. In the rest of this chapter, we use $\alpha = (N_\theta N_d)^{-1}$.

To find an approximation method for the SIRT method, we start by rewriting the equation of a single SIRT iteration (Eq. (4.6)) in a matrix format:

$$\boldsymbol{x}_s^{k+1} = (\boldsymbol{I} - \alpha \boldsymbol{W}^T \boldsymbol{W})\boldsymbol{x}_s^k + \alpha \boldsymbol{W}^T \boldsymbol{p} \tag{4.11}$$

This is a recursion equation of the form $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{b}$, which has the following solution for iteration n:

$$\boldsymbol{x}_s^n = \boldsymbol{A}^n \boldsymbol{x}_s^0 + \alpha \left[\sum_{k=0}^{n-1} \boldsymbol{A}^k\right] \boldsymbol{W}^T \boldsymbol{p} \tag{4.12}$$

where $\boldsymbol{A} = \boldsymbol{I} - \alpha \boldsymbol{W}^T \boldsymbol{W}$. Often, the initial image of the SIRT method is set to the zero image ($\boldsymbol{x}_s^0 = \boldsymbol{0}$), in which case we end up with:

$$\boldsymbol{x}_s^n = \alpha \left[\sum_{k=0}^{n-1} \boldsymbol{A}^k\right] \boldsymbol{W}^T \boldsymbol{p} \tag{4.13}$$

Now, we want to find a method that can approximate Eq. (4.13). In order to find such a method, we look at the FBP method, and note that, in parallel-beam tomography, convolving a sinogram with a filter and backprojecting the result is identical to backprojecting the sinogram and convolving the resulting image with the backprojected filter:

$$FBP(\boldsymbol{p}, \boldsymbol{h}) = \boldsymbol{H}_{h'} \boldsymbol{W}^T \boldsymbol{p} \tag{4.14}$$

Algorithm 4.1 Compute an FBP filter that approximates n iterations of SIRT

```
Require: $W \in \mathbb{R}^{N_d N_\theta \times N^2}$, $n \in \mathbb{Z}^+$, $\alpha \in \mathbb{R}$
  $q_0 \leftarrow 0$
  $c \leftarrow [0, \dots, 0, 1, 0, \dots, 0]^T$
  for $k = 1$ to $n$ do
    $q_k \leftarrow q_{k-1} + c$
    $c \leftarrow c - \alpha W^T W c$
  end for
  $u_n \leftarrow \alpha W q_n$
  return $u_n$
```

Here, $\boldsymbol{H}_{\boldsymbol{q}}$ is a 2D convolution with filter $\boldsymbol{q}$, and $\boldsymbol{h} = \boldsymbol{W}\boldsymbol{h}'$.

Note the similarities between the rewritten SIRT equation (Eq. (4.13)) and the rewritten FBP equation (Eq. (4.14)), which suggest that we can approximate the SIRT equation by approximating $\sum_{k=0}^{n-1} \boldsymbol{A}^k$ by a 2D convolution operation with filter $\boldsymbol{q}_n$:

$$\boldsymbol{x}_s^n \approx \alpha \boldsymbol{H}_{\boldsymbol{q}_n} \boldsymbol{W}^T \boldsymbol{p} \tag{4.15}$$

A good approximating filter $\boldsymbol{q}_n$ can be found by taking the impulse response of $\sum_{k=0}^{n-1} \boldsymbol{A}^k$:

$$\boldsymbol{q}_n = \sum_{k=0}^{n-1} \boldsymbol{A}^k [0, \dots, 0, 1, 0, \dots, 0]^T \tag{4.16}$$

In other words, we apply $\boldsymbol{A}$ to an image $n-1$ times, starting with an image with only the central pixel set to 1, and sum the resulting images to obtain the 2D filter $\boldsymbol{q}_n$.

Since backprojecting a sinogram and convolving the resulting image is the same as convolving the sinogram with the forward projected filter and backprojecting the result, we can write this as:

$$\begin{aligned} \boldsymbol{x}_s^n &\approx \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{u}_n} \boldsymbol{p} \\ \boldsymbol{u}_n &= \alpha \boldsymbol{W} \boldsymbol{q}_n \end{aligned} \tag{4.17}$$

Here, $\boldsymbol{C}_{\boldsymbol{h}}$ is the same convolution operator as in Eq. (4.3), and $\boldsymbol{u}_n$ is the corresponding angle-dependent filter. Comparing Eq. (4.3) and Eq. (4.17), we conclude that the SIRT method with n iterations can be approximated by the FBP method with a special filter $\boldsymbol{u}_n$:

$$\boldsymbol{x}_s^n \approx FBP(\boldsymbol{p}, \boldsymbol{u}_n) \tag{4.18}$$

To summarize, the algorithm to compute an approximating filter is given in Algorithm 4.1. For more information on implementing this method, and results for non-local tomographic reconstruction, we refer to Chapter 3.

One advantage of this approximation is that, after calculating the filter, the final reconstruction method is identical to standard FBP. Therefore, we can use the same approach as for FBP to evaluate it locally: simply exchanging $\boldsymbol{W}^T$ with $\boldsymbol{W}_{\mathscr{L}}^T$:

$$\boldsymbol{x}_s^n \approx FBP_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{u}_n) \tag{4.19}$$

Results for locally approximating SIRT with this approach are given in Section 4.5.

Including regularization

As explained in Section 4.2, many regularized iterative methods include a SIRT step in their iterative equations. In Section 4.3, we showed that we can approximate these SIRT steps locally by using the proposed filter method. However, to locally approximate the complete regularized iterative methods, we need to perform some extra steps. We start by explicitly splitting the reconstruction image at iteration k into two parts: a standard SIRT image $\boldsymbol{x}_s^k$ and a prior-based correction term $\boldsymbol{y}^k$:

$$\boldsymbol{x}^k = \boldsymbol{x}_s^k + \boldsymbol{y}^k \tag{4.20}$$

Furthermore, we rewrite the equation for a single iteration of these methods, such that it consists of a single SIRT step on the previous iteration, and an additional correction term $\boldsymbol{d}$ that incorporates the prior knowledge:

$$\boldsymbol{x}^{k+1} = S(\boldsymbol{x}^k) + \boldsymbol{d}^{k+1} \tag{4.21}$$

Note that it is usually straightforward to rewrite a regularized iterative method that uses SIRT to this form, although one would typically not use such a formulation in practice. For example, SIRT with box constraints (Eq. (4.8)) can be written in this form by taking:

$$\boldsymbol{d}_i^{k+1} = \begin{cases} l - S(\boldsymbol{x}^k)_i & : \text{ if } S(\boldsymbol{x}^k)_i < l \\ r - S(\boldsymbol{x}^k)_i & : \text{ if } S(\boldsymbol{x}^k)_i > r \\ 0 & : \text{ otherwise} \end{cases} \tag{4.22}$$

As another example, iterations of the ISTA method with a wavelet basis (Eq. (4.9)) can be written in the form of Eq. (4.21) by taking:

$$\boldsymbol{d}^{k+1} = \boldsymbol{B}^{-1}\mathscr{P}_\lambda(\boldsymbol{B}S(\boldsymbol{x}^k)) - S(\boldsymbol{x}^k) \tag{4.23}$$

Now, we aim to find a local approximation to Eq. (4.21). If we apply a single SIRT iteration to $\boldsymbol{x}^k$, we get:

$$\begin{aligned} S(\boldsymbol{x}^k) &= \boldsymbol{A}\left(\boldsymbol{x}_s^k + \boldsymbol{y}^k\right) + \alpha \boldsymbol{W}^T\boldsymbol{p} \\ &= \boldsymbol{A}\boldsymbol{x}_s^k + \alpha \boldsymbol{W}^T\boldsymbol{p} + \boldsymbol{A}\boldsymbol{y}^k \\ &= S(\boldsymbol{x}_s^k) + \boldsymbol{A}\boldsymbol{y}^k \end{aligned} \tag{4.24}$$

By combining Eq. (4.21) and Eq. (4.24), we see that:

$$\boldsymbol{x}^{k+1} = S(\boldsymbol{x}_s^k) + \boldsymbol{A}\boldsymbol{y}^k + \boldsymbol{d}^{k+1} \tag{4.25}$$

Using the definition of Eq. (4.20), we can take:

$$\begin{aligned} \boldsymbol{x}_s^{k+1} &= S(\boldsymbol{x}_s^k) \\ \boldsymbol{y}^{k+1} &= \boldsymbol{A}\boldsymbol{y}^k + \boldsymbol{d}^{k+1} \end{aligned} \tag{4.26}$$

In order to locally approximate Eq. (4.21), we need to find local approximations for $\boldsymbol{x}_s^{k+1}$ and $\boldsymbol{y}^{k+1}$.

The iterations of $\boldsymbol{x}_s^{k+1}$ are identical to SIRT iterations, for which we already derived a local approximation in Section 4.3:

$$\boldsymbol{x}_s^{k+1} \approx FBP_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{u}_{k+1}) \tag{4.27}$$

Furthermore, we can choose to only apply the prior knowledge inside the local part $\mathscr{L}$. In this case, the prior-based correction term $\boldsymbol{d}^{k+1}$ is only nonzero for pixels inside $\mathscr{L}$. To find a local approximation to $\boldsymbol{A}\boldsymbol{y}^k$, we expand $\boldsymbol{A}$, and use the definition of the local and outer projection operations Eq. (4.2):

$$\begin{aligned}\boldsymbol{A}\boldsymbol{y}^k &= \left(\boldsymbol{I} - \alpha \boldsymbol{W}^T \boldsymbol{W}\right)\boldsymbol{y}^k \\ &= \boldsymbol{y}^k - \alpha\left(\boldsymbol{W}_{\mathscr{L}}^T + \boldsymbol{W}_{\mathscr{F}}^T\right)\boldsymbol{W}\boldsymbol{y}^k \\ &= \boldsymbol{y}^k - \alpha \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{W}\boldsymbol{y}^k - \alpha \boldsymbol{W}_{\mathscr{F}}^T \boldsymbol{W}\boldsymbol{y}^k\end{aligned} \tag{4.28}$$

We approximate Eq. (4.28) locally by simply ignoring the term $\alpha \boldsymbol{W}_{\mathscr{F}}^T \boldsymbol{W}\boldsymbol{y}^k$ which affects the pixels outside $\mathscr{L}$. By ignoring this term, we ignore the effect that the local prior has on the pixels outside $\mathscr{L}$, which can affect the pixels inside $\mathscr{L}$ in later iterations. Since we are, in the end, only interested in the reconstruction inside $\mathscr{L}$, this approximation is usually sufficiently accurate in practice. Another result of this approximation is that $\boldsymbol{y}^k$ will be zero outside $\mathscr{L}$ for any iteration k, and therefore we can substitute $\boldsymbol{W}_{\mathscr{L}}$ for $\boldsymbol{W}$ in the forward projection as well:

$$\boldsymbol{A}\boldsymbol{y}^k \approx \boldsymbol{y}^k - \alpha \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{W}_{\mathscr{L}} \boldsymbol{y}^k \tag{4.29}$$

To summarize, we have derived a method to approximate a regularized iterative method inside $\mathscr{L}$. Starting with $\boldsymbol{y}^0 = \boldsymbol{0}$, we use the following iterations:

$$\begin{aligned}\boldsymbol{x}_s^{k+1} &= FBP_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{u}_{k+1}) = \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{C}_{\boldsymbol{u}_{k+1}} \boldsymbol{p} \\ \boldsymbol{y}^{k+1} &= \boldsymbol{y}^k - \alpha \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{W}_{\mathscr{L}} \boldsymbol{y}^k + \boldsymbol{d}^{k+1} \\ \boldsymbol{x}^{k+1} &= \boldsymbol{x}_s^{k+1} + \boldsymbol{y}^{k+1}\end{aligned} \tag{4.30}$$

Note that every projection operation in Eq. (4.30) is local, and can therefore be computed efficiently. The needed filters $\boldsymbol{u}_k$ for all iterations can be precomputed for a certain projection geometry with a single run of Algorithm 4.1 by returning a filter for each iteration. The method is based on three approximations to a standard regularized iterative method:

1. Iterations of SIRT are approximated by FBP with specific filters.

2. The prior knowledge is only applied inside $\mathscr{L}$.

3. The effect of the local prior on pixels outside $\mathscr{L}$ is ignored.

Results from Section 4.5 will show that despite these approximations, reconstructions computed by our method are of significantly higher quality than either local FBP or global SIRT reconstructions, and visually similar to global regularized iterative reconstructions. The method is summarized in Algorithm 4.2.

Algorithm 4.2 Compute a local approximation to a regularized iterative method

```
Require: $\boldsymbol{p} \in \mathbb{R}^{N_d N_\theta}$, $\boldsymbol{W} \in \mathbb{R}^{N_d N_\theta \times N^2}$, $n \in \mathbb{Z}^+$, $\alpha \in \mathbb{R}$
  $\boldsymbol{y}^0 \leftarrow \boldsymbol{0}$
  for $k = 1$ to $n$ do
      $\boldsymbol{x}_s^k \leftarrow FBP_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{u}_k)$
      $\boldsymbol{y}^k \leftarrow \boldsymbol{y}^{k-1} - \alpha \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{W}_{\mathscr{L}} \boldsymbol{y}^{k-1} + \boldsymbol{d}^k$
  end for
  return $\boldsymbol{x}_s^n + \boldsymbol{y}^n$
```

Algorithm 4.3 Compute a local approximation to FISTA minimizing $\|\nabla \boldsymbol{x}\|_1$

```
Require: $\boldsymbol{p} \in \mathbb{R}^{N_d N_\theta}$, $\boldsymbol{W} \in \mathbb{R}^{N_d N_\theta \times N^2}$, $n \in \mathbb{Z}^+$, $n_{FGP} \in \mathbb{Z}^+$, $\alpha \in \mathbb{R}$
  $t^0 \leftarrow 1$
  $\boldsymbol{x}_{\mathscr{L}}^0 \leftarrow \boldsymbol{0}$
  $\boldsymbol{x}^0 \leftarrow \boldsymbol{0}$
  for $k = 1$ to $n$ do
      $\boldsymbol{x}_s \leftarrow FBP_{\mathscr{L}}(\boldsymbol{p}, \boldsymbol{u}_k)$
      $\boldsymbol{q} \leftarrow \boldsymbol{x}_{\mathscr{L}}^{k-1} - \alpha \boldsymbol{W}_{\mathscr{L}}^T \boldsymbol{W}_{\mathscr{L}} \boldsymbol{x}_{\mathscr{L}}^{k-1}$
      $\boldsymbol{x}^k \leftarrow FGP(\boldsymbol{x}_s + \boldsymbol{q}, n_{FGP})$
      $t^k \leftarrow (1 + \sqrt{1 + 4t^{k-1}})/2$
      $\boldsymbol{r} \leftarrow \boldsymbol{x} + (t^{k-1} - 1)\boldsymbol{x}^k/(t^k \boldsymbol{x}^{k-1})$
      $\boldsymbol{x}_{\mathscr{L}}^k \leftarrow \boldsymbol{r} - \boldsymbol{x}_s^k$
  end for
  return $\boldsymbol{x}^n$
```

The term $\boldsymbol{d}$ in Algorithm 4.2 is the term in which the prior knowledge is exploited, and depends on which regularized iterative method is used. Often, in actual implementations, a different formulation can be used that is more natural to that specific regularized iterative method than the one shown in Algorithm 4.2. As an example, Algorithm 4.3 shows an implementation of the method when using FISTA to minimize the ℓ_1 norm of the gradient of the reconstructed image. Here, we use similar notation to [BT09b], and $FGP(\boldsymbol{x}, n_{FGP})$ refers to the FGP method of [BT09b] with n_{FGP} iterations, applied to the image $\boldsymbol{x}$.

Implementation details

In this section, we will discuss a few details on implementing the proposed method. Specifically, we will discuss how to prevent certain reconstruction artifacts from appearing and how to improve the computation time of the method in repeated applications.

Using some forms of prior knowledge, artifacts can appear in the reconstructed image near the edges of the reconstruction grid. For example, the gradient in a total variation constraint is often defined differently for pixels on the edge of the reconstruction grid compared to pixels in the interior, which can affect the reconstruction near the edges. For global regularized iterative methods, the interesting features of the reconstructed object are usually situated relatively far from the edge, in which case the artifacts near edges can simply be ignored. In the proposed local method, however, interesting features may be located near or on the edge of the chosen local part. A simple but effective way of reducing the effect of edge artifacts in these cases is to increase the size of the local part slightly, and crop the resulting reconstruction to the chosen local part. In the rest of this chapter, we increase the size of the local part by padding it with $\frac{1}{8}$ of the height/width of the local part on each side.

The reconstruction quality of the filter-based approximation of the SIRT method given in Section 4.3 depends on the discrete implementations of the projection operators, as explained in Chapter 3. Specifically, the method is based on approximating the combined $\boldsymbol{W}^T\boldsymbol{W}$ operator by a shift-invariant convolution operation. The discrete projection operations can be implemented in different ways [XM06], and the accuracy of the approximation depends on the chosen implementation. In practice, most artifacts resulting from the errors in the approximation are found in the low frequencies of the reconstructed image, similar to the artifacts that can occur when discretizing the Ram-Lak filter of the FBP method [KS01, Fig. 3.13]. By using implementations of the projection operators that minimize the approximation error that is made, reconstruction artifacts can be limited, and are typically invisible to a human observer. In this chapter, we use an additional preprocessing step to further reduce these artifacts. Before each reconstruction with the local approximation method, we subtract from the projection data the forward projection of a disc, centered on the rotation axis, with a diameter N and a constant gray value. The gray value is chosen such that the ℓ_2-norm of the zero-frequency components of all projections are minimized after subtraction. By reducing the low-frequency components of the projection data with this procedure, the artifacts resulting from the approximation error are reduced as well. After reconstruction, the same disc is added back to the reconstructed image. In practice, this procedure ensures that artifacts resulting from errors made in approximating SIRT by filtered backprojection are minimal.

As explained in Section 4.3, all projection operations of the proposed method can be computed locally, and are therefore efficient to compute. When the local part is much smaller than the number of detector pixels ($N_{\mathscr{L}} \ll N_d$), however, the convolution operation in $FBP_{\mathscr{L}}$, which scales with N_d instead of $N_{\mathscr{L}}$, can become a significant part of the total computation time. In many cases, however, one will perform repeated applications of the local method, for example when finding optimal parameters for the applied prior knowledge term, or when reconstructing multiple local parts at different locations. In these cases, the convolution of the projection data with the different filters $\boldsymbol{u}_k$ for each iteration can be precomputed once and reused for the different local reconstructions, improving reconstruction time significantly.

4.4 Experiments

To investigate the properties of the proposed method, we implemented it in Python, version 3.4.3, using the ASTRA toolbox [Aar+15] to perform all tomographic projection operations, which enables the use of optimized GPU-based computations [PBS11]. All experiments were performed on a machine running Fedora Linux 21, with an Intel Xeon E5-2623 processor, 13 GB of memory, and a NVIDIA GeForce GTX TITAN Z GPU using CUDA version 7.0.

We present results for three different forms of prior knowledge about the reconstructed object: one domain constraint and two penalty constraints. For the domain constraint we use box constraints on the pixel values by specifying $D = \{\boldsymbol{x} \in \mathbb{R}^{N^2}; l \leq x_i \leq r,\ i = 1, \ldots, N^2\}$ in the objective function of Eq. (4.7). For the penalty constraints, we use ℓ_1 minimization of the reconstruction in a Haar wavelet basis, i.e. specifying $g(\boldsymbol{x}) = \|\boldsymbol{B}\boldsymbol{x}\|_1$, and ℓ_1 minimization of the gradient of the reconstructed image (TV minimization), i.e. specifying $g(\boldsymbol{x}) = \|\nabla \boldsymbol{x}\|_1$. We use Eq. (4.8) to find solutions in the case of box constraints on the pixel values, and the FISTA method in the case of both ℓ_1 penalty functions. In all cases, we compare the locally approximated reconstructions with global reconstructions of the full object exploiting the same prior knowledge on the full volume, and with the popular analytical FBP method and algebraic SIRT method, which do not explicitly exploit any prior knowledge.

The phantom that is used in most experiments in this chapter is shown in Fig. 4.3. This phantom was chosen because it is suitable for all three forms of prior knowledge that we exploit. It consists of two materials: a background with a value of zero and a foreground with a value of one. Therefore, box constraints can be effectively exploited by setting $l = 0$ and $r = 1$. Since the phantom has a sparse boundary, TV minimization and a Haar wavelet basis can also be used to improve reconstruction quality. In addition to the phantom shown in Fig. 4.3, we also present some results for the Shepp-Logan head phantom, shown in Fig. 4.9a, which has a relatively sparse boundary as well.

For each reconstruction, we report the mean squared error (MSE) of the reconstructions inside the region of interest, compared to a known ground truth image. We also report the structural similarity index (SSIM) [Wan+04] of the reconstructions inside the region of interest compared to the ground truth, which is a metric that is designed to be closer to the human visual system than the mean squared error. For methods where a parameter needs to be chosen, i.e. λ in Eq. (4.7), we perform two reconstructions each time: one with the value that minimizes the *MSE* and one with the value that maximizes the *SSIM*. In each case, we find the optimal parameter value using the Nelder-Mead method [NM65]. Note that the optimal parameter value can depend on the dimensions of the reconstruction grid, and therefore, the optimal values can be different for the global regularized iterative reconstructions compared to the locally approximated reconstructions. For all iterative methods, we use 200 iterations to compute each reconstruction, and we use 100 FGP iterations in the FISTA method for TV minimization [BT09b].

In most experiments, we use a 4096 × 4096 pixel image of the phantom, and generate projection data for 4096 detector pixels. Afterwards, the projection data is

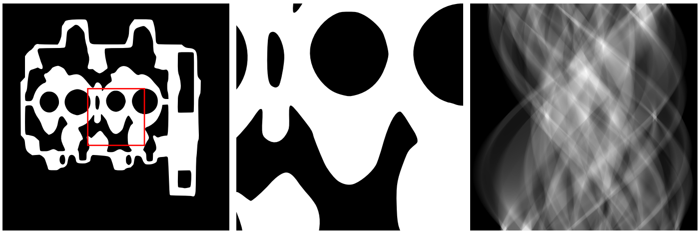

(a) (b) (c)

Figure 4.3: The phantom used for most experiments in this chapter. In (a), the entire phantom is shown with a red square indicating the local part (b) that is used in most experiments. In (c), the sinogram of the phantom is shown for 1024 detector pixels and 1024 projections equally distributed in $[0, \pi]$.

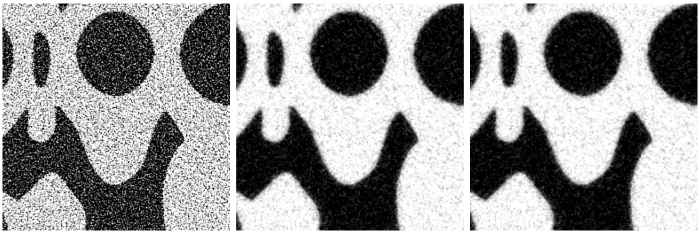
resampled to 1024 detector pixels, and reconstructions are computed on a 1024×1024 pixel grid, or a local part of that grid. These reconstructions are compared to the original 4096×4096 pixel phantom, resampled to a 1024×1024 pixel grid. In most cases, additional Poisson noise is applied to the projection data to simulate experimental conditions. The amount of applied Poisson noise is indicated by a variable I_0, with lower values corresponding to higher amounts of applied noise. Specifically, the noise is applied by first transforming the simulated projections to virtual photon counts, in which the largest photon count out of all detector pixels is set to I_0. For each detector pixel, a new photon count is sampled from a Poisson distribution with the original photon count as the expected value. Finally, the resulting noisy photon counts are transformed back to noisy line integrals of the phantom.

4.5 Results

In this section, we present the results of the experiments that we performed to investigate the properties of the proposed local approximation method, and discuss these results.

Local SIRT approximation

In Fig. 4.4, reconstructions are shown for the local part of the phantom, computed by standard FBP, standard SIRT, and the local approximation of SIRT (Eq. (4.19)). Note that the global SIRT reconstruction and its local approximation are visually very similar. The difference between the computation times is significant, however: the local reconstructions take 28 milliseconds to compute each, while the global SIRT reconstruction takes 2.6 seconds. The *MSE* of the FBP, SIRT, and local approximation are 0.245, 0.016, and 0.016, respectively, and the *SSIM* values are 0.07, 0.25, and 0.27.



(a) (b) (c)

Figure 4.4: Reconstructions of a 256 × 256 pixel local part of the motor phantom (Fig. 4.3a), using projection data of 1024 detector pixels with $N_\theta = 512$ projection angles, equally distributed in the interval $[0, \pi]$, and with Poisson noise applied. In (a) the local FBP reconstruction is shown, in (b) the global SIRT reconstruction cropped to the local part, and in (c) the locally approximated SIRT reconstruction.

Local regularized iterative approximation

In Fig. 4.5, the mean squared error and structural similarity index are shown as a function of the amount of applied Poisson noise I_0, for standard FBP, standard SIRT, and global and locally approximated reconstructions using various types of prior knowledge. The results show that by exploiting prior knowledge, reconstruction quality can be significantly improved compared to standard FBP and SIRT reconstructions. For this phantom, exploiting total variation minimization yields reconstructions with the lowest *MSE* and highest *SSIM* values. The results also show that for all tested types of prior knowledge, the quality metrics of the locally approximated reconstructions are very close to those of the global regularized iterative reconstructions. For unknown reasons, the quality metrics of the local approximations are slightly better than the global regularized iterative reconstructions. Similar results can be seen in Fig. 4.6, where the quality metrics are shown as a function of the number of projections angles.

The mean squared error and structural similarity index are shown as a function of the size of the local part $\mathscr{L}$ in Fig. 4.7. For all three prior knowledge types, the reconstruction quality of the local approximations is only significantly lower compared to the global regularized iterative methods when the local size is $N_{\mathscr{L}} = 32$ or smaller, at which point the number of pixels of the local part is less than 0.1% of the number of pixels in the global reconstruction grid. For larger local sizes, the reconstruction quality is almost independent of the local size. These results suggest that, even for reasonably small local parts, the approximations that are made by the proposed local method do not influence the reconstruction quality significantly.

Reconstructed images of a local part with 256 × 256 pixels are shown in Fig. 4.8, for projection data of 1024 detector pixels and 512 equiangular projections with Poisson noise applied. The images show that the local approximations are visually almost identical to the global regularized iterative reconstructions for all three prior knowledge types. The results also show how the different prior knowledge types can help improve certain image characteristics compared to standard FBP and SIRT reconstructions. In Fig. 4.9, reconstructed images are shown for a smaller local part (128 × 128 pixels) of the Shepp-Logan head phantom. Similar to the previous results,

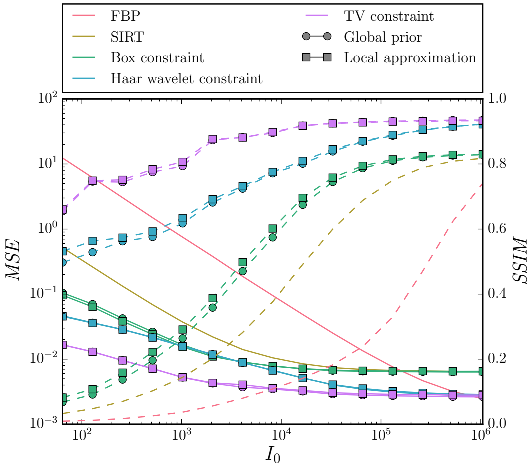


Figure 4.5: Mean squared error (*MSE*, solid lines) and structural similarity index (*SSIM*, dashed lines) of reconstructions of a region (256 × 256 pixels) of the motor phantom (Fig. 4.3a) for various amounts of applied Poisson noise I_0 and types of prior knowledge. The reconstructions are computed using projection data of 1024 detector pixels and 512 projections equally distributed in the interval $[0, \pi]$.

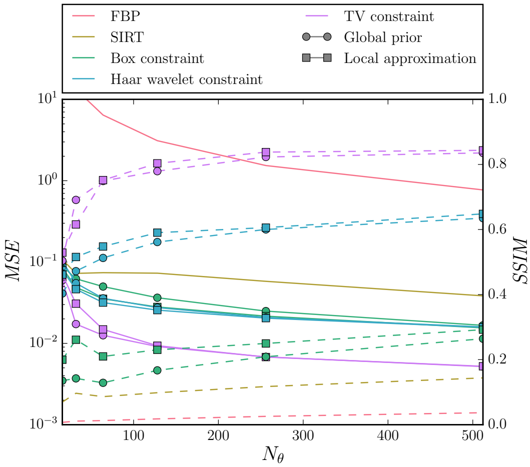


Figure 4.6: Mean squared error (*MSE*, solid lines) and structural similarity index (*SSIM*, dashed lines) of reconstructions of a region (256 × 256 pixels) of the motor phantom (Fig. 4.3a) for various numbers of projection angles N_θ (equally distributed in the interval $[0, \pi]$) and types of prior knowledge. The reconstructions are computed using projection data of 1024 detector pixels, with applied Poisson noise.

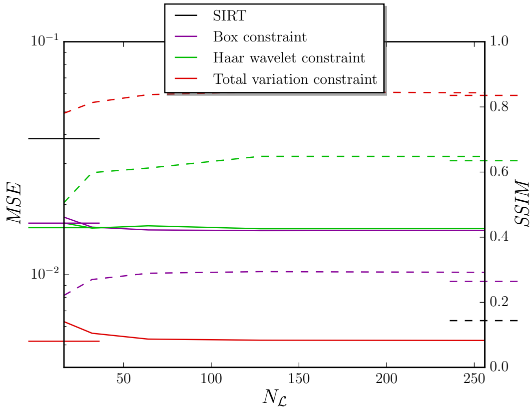


Figure 4.7: Mean squared error (*MSE*, solid lines) and structural similarity index (*SSIM*, dashed lines) of reconstructions of a region (256 × 256 pixels) of the motor phantom (Fig. 4.3a) for various sizes of the local part $N_{\mathscr{L}}$ and types of prior knowledge. The reconstructions are computed using projection data of 1024 detector pixels and 512 projections equally distributed in the interval $[0, \pi]$, with applied Poisson noise. For $N_{\mathscr{L}} < 256$, multiple local reconstructions are tiled to create a reconstruction of 256 × 256 pixels, to enable comparison between different local sizes. The partial horizontal lines on each axis indicate the *MSE* and *SSIM* of global SIRT and global regularized iterative reconstructions, cropped to the same 256 × 256 pixels.

the local approximations are visually almost identical to the global regularized iterative reconstructions.

Computation time

The computation time of the proposed local reconstruction method is shown in Fig. 4.10 as a function of the size of the local part $\mathscr{L}$. Also shown is the computation time of the standard global regularized iterative method. For the local method, computation times are shown both for the first application, as well as for subsequent applications, in which the convolution results of the first application can be reused to decrease the needed computation time (see Section 4.3). For all of types prior knowledge, the local method requires significantly less computation time than the global regularized iterative methods.

If one is only interested in a local part of the object, the local method can be used to compute advanced regularized reconstructions in a few seconds instead of the several minutes it costs to compute the global reconstruction. In cases where the same regularized iterative method is computed multiple times for the same projection data, for example when estimating the λ parameter, the proposed local method requires even less computation time, leading to a significant reduction of processing time in practice. Finally, since each local reconstruction is independent of the other local reconstructions, different local parts can be reconstructed in parallel and combined afterwards to compute a larger part of the scanned object in short time. An example of such a reconstruction is shown in Section 4.5.

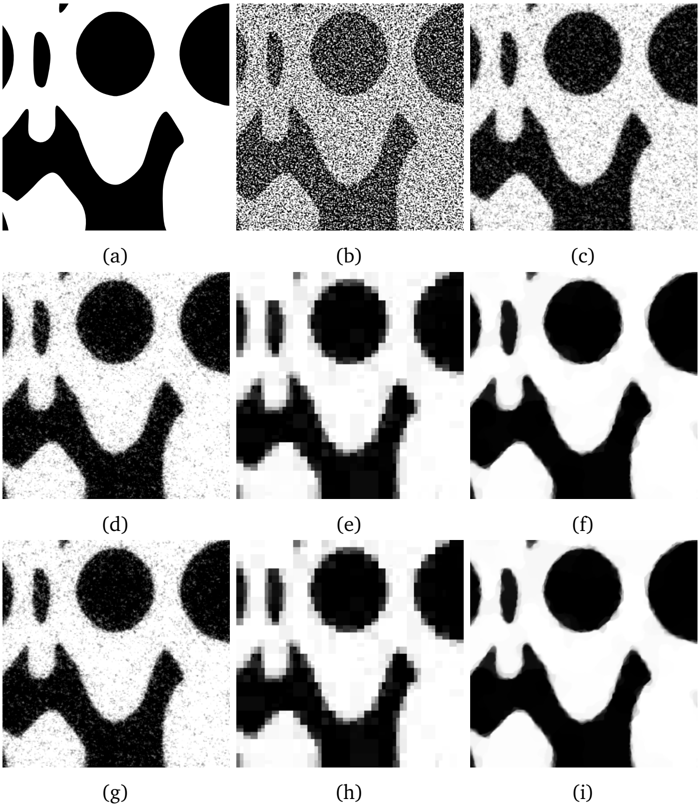


Figure 4.8: Reconstructions of a local part (256 × 256 pixels) of the motor phantom (a) from projection data of 1024 detector pixels and 512 projections equally distributed in the interval $[0, \pi]$, with Poisson noise applied, using various reconstruction methods: (b) local FBP, (c) global SIRT cropped to local part, (d)-(f) global regularized iterative method cropped to local part, with (d) box constraint, (e) Haar wavelet constraint, and (f) TV constraint, and (g)-(i) the proposed local method, with (g) box constraint, (h) Haar wavelet constraint, and (i) TV constraint. The local reconstructions are shown with a gray-level window in which black corresponds to the minimum value and white to the maximum value of the phantom inside the local part.

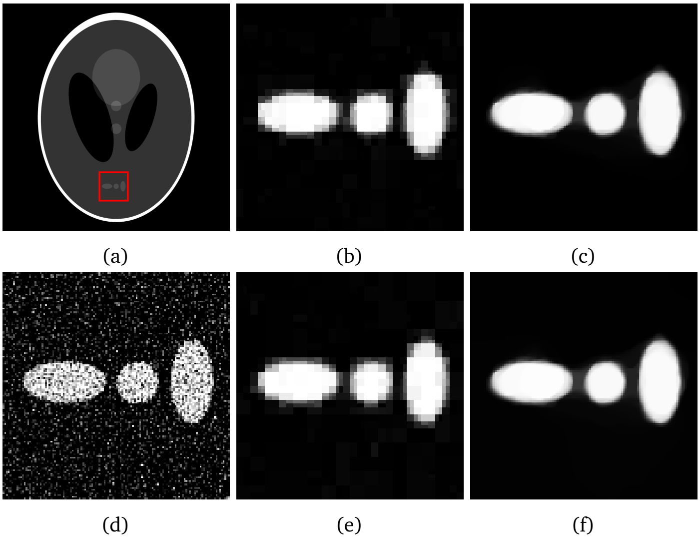


Figure 4.9: Reconstructions of a local part (128 × 128 pixels) of the Shepp-Logan head phantom, indicated by the red square in (a). The reconstructions are computed from projection data of 1024 detector pixels and 512 projections equally distributed in the interval $[0, \pi]$, with Poisson noise applied, using various reconstruction methods: (b)-(c) global regularized iterative method cropped to local part, with (b) Haar wavelet constraint and (c) TV constraint, (d) local FBP, and (e)-(f) the proposed local method, with (e) Haar wavelet constraint and (f) TV constraint. The local reconstructions are shown with a gray-level window in which black corresponds to the minimum value and white to the maximum value of the phantom inside the local part.

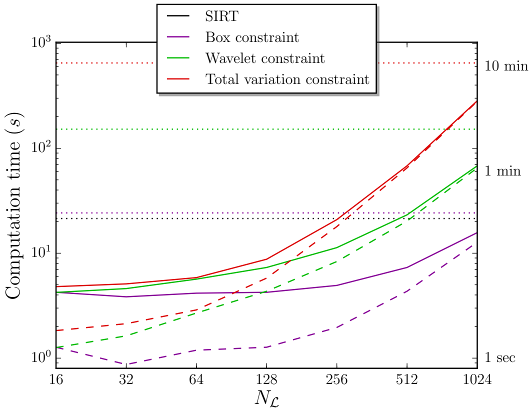


Figure 4.10: Reconstruction time of the global regularized iterative methods (dotted) and the proposed local method for various sizes of the local part $N_\mathcal{L}$ and constraint types, using data of 2048 detector pixels and 512 projections. Solid lines show the reconstruction time for a single application of the local method, and dashed lines show the reconstruction time for subsequent applications, where the convolution results of an earlier reconstruction can be reused (see Section 4.3).

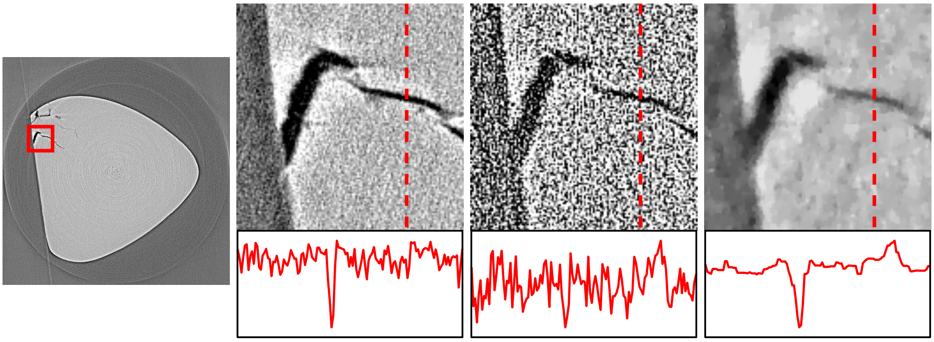
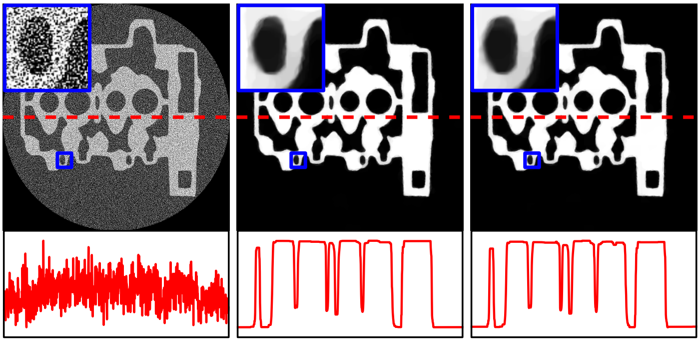

(a) (b) (c) (d)

Figure 4.11: Reconstructions of a local part (128 × 128 pixels) of experimental data of a small fatigue test sample made from Ti alloy VST 55531, acquired with 1200 detector pixels and 1500 projections equally distributed over the interval $[0, \pi]$. In (a) and (b), FBP reconstructions are shown using all 1500 projections, with the local part indicated by a red square in (a). The local FBP reconstruction using only 75 equiangular projections is shown in (c). In (d), a reconstruction is shown for the same 75 projections, using the local reconstruction method presented in this chapter with TV-minimization regularization by the FISTA method. Underneath each local reconstruction, the line profile of the column indicated by the dashed line is shown.

Experimental data

In Fig. 4.11, reconstructed images are shown for a local part of an experimental dataset. The experimental data was acquired for a small fatigue test sample made from Ti alloy VST 55531. The sample was scanned at beamline ID11 of the European Synchrotron Radiation Facility (ESRF), with a parallel, monochromatic (52 keV) synchrotron X-ray beam. The distance between the sample and detector was 40 mm, and 1500 projections were acquired, equally distributed in the interval $[0, \pi]$. The projections were acquired on a high resolution detector system, resulting in projections, after 2 × 2 binning, with 1200 × 1200 pixels and an effective pixel size of 0.56 microns.

Results are shown in Fig. 4.11 for a single slice of the reconstructed dataset, computed using FBP and the proposed local method with a TV minimization constraint. For FBP, we show results both when using all 1500 projections that were acquired, and when using only 75 projections, selected by taking every 20$^{\text{th}}$ projection of the full dataset. For the local method, we show results for the same limited dataset of 75 projections. The results show that the local method can be successfully applied to an experimental dataset to exploit prior knowledge in the reconstruction. Compared to the FBP reconstruction using 75 projections, the local method is able to more clearly separate the formed crack from the sample itself, which is especially visible in the line profiles. Note that in this type of sample, a user would typically only be interested in the highly localized crack that is forming in the sample, which would make global regularized iterative methods waste significant amounts of computation time on parts of the sample that are not interesting. With the proposed local method, on the other hand, a user would be able to select and reconstruct only those parts of the sample that are interesting.



(a) (b) (c)

Figure 4.12: Reconstructions of the motor phantom using projection data of 1024 detectors and $N_\theta = 512$ projection angles, equally distributed in the interval $[0, \pi]$, with Poisson noise applied. The reconstructions are computed with (a) FBP, (b) global TV minimization by the FISTA method, and (c) local 128 × 128 pixel reconstructions tiled to the complete 1024 × 1024 pixel grid. The local reconstructions in (c) are computed using the local reconstruction method presented in this chapter with TV-minimization regularization by the FISTA method. Underneath each reconstruction, the line profile of the row indicated by the dashed line is shown. A small region, indicated by the blue square, is shown enlarged in the top-left corner of each reconstruction as well.

Tiling reconstructions

As explained before, one possibility of the proposed local method is to reconstruct different local parts of the image and combine them afterwards into a single reconstruction. One application of this approach would be to compute the different local parts in parallel, which can be parallelized efficiently since each local reconstruction is independent of the others. Another application would be to estimate reconstruction parameters such as the λ term of Eq. (4.7) only in a local part of the reconstruction, which would significantly reduce the time needed to estimate them. Afterwards, the complete image can be reconstructed by combining several local reconstructions using these parameters, which can be computed in parallel as well.

An example of a reconstruction that is computed by tiling several local reconstructions is shown in Fig. 4.12. In this case, we combined 64 local reconstructions of 128 × 128 pixels each to compute a single 1024 × 1024 pixel reconstruction, using TV-minimization as the prior knowledge term. The local reconstructions are tiled by simply placing them next to each other on the large reconstruction grid, without any overlapping regions. The results show that there are no visible artifacts from this tiling procedure. Furthermore, the tiled reconstruction is visually almost identical to a reconstruction computed by the global regularized iterative method. This shows that it is possible to significantly reduce the computation time of a global regularized iterative reconstruction method by approximating it with a tiling of local reconstructions computed in parallel.

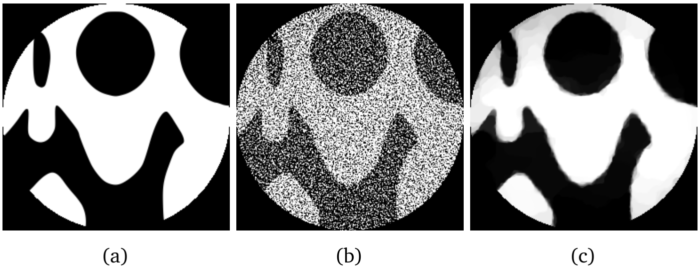

(a) (b) (c)

Figure 4.13: Reconstructions of the motor phantom (a), using projection data of 1024 detectors truncated to the central 256 detector pixels, using $N_\theta = 512$ projection angles, equally distributed in the interval $[0, \pi]$, and with Poisson noise applied. The reconstructions of FBP (b) and the proposed method with TV-minimization by the FISTA method (c) are shown for the central disc with a width of 256 pixels. Constant padding is used in both reconstructions to reduce truncation artifacts.

Truncated projection data

In some applications of tomography, it is impossible to acquire projections that include the entire scanned object. In these cases, the acquired projection data are *truncated* at the edge of the detector. The resulting reconstruction problem is similar to local reconstruction: again, one is only interested in a subvolume of the entire scanned object. In this case, however, data for the object outside the subvolume is missing. Filtered backprojection is often used to reconstruct truncated data by simply padding the acquired data in order to reduce the artifacts caused by the truncation. Since the local method proposed in this chapter uses FBP to approximate the SIRT method, the same padding approach can be used to apply the method to truncated data. Reconstructions of truncated phantom data are shown in Fig. 4.13, for FBP and the proposed local method. The results show that the local method can be used to exploit prior knowledge in the case of truncated data to improve reconstruction quality.

4.6 Conclusions

In this chapter, we introduced a method to approximate regularized iterative tomographic reconstruction methods inside a region of interest. This method can be used to reduce computation time when one is only interested in the reconstruction inside the region of interest. The method is based on approximating the SIRT steps that are part of many regularized iterative methods by filtered backprojection with specific pre-calculated filters. The result is a reconstruction method in which all projection operations involve only the pixels that are inside the region of interest. The method can also be applied to truncated projection data by similar padding techniques as used for filtered backprojection.

To investigate the properties of the proposed method, we computed reconstructions using various types of prior knowledge about the reconstructed object: box constraints on the pixel values, ℓ_1 minimization of the reconstruction in a wavelet basis, and ℓ_1

minimization of the gradient of the reconstructed image. The results show that the proposed method is able to accurately approximate the reconstructions that would be the result of computing the regularized iterative methods on the full object. Compared to standard reconstruction methods such as FBP and SIRT, the proposed method is able to significantly improve reconstruction quality by exploiting prior knowledge.

One interesting application of the method is to use it to tile reconstructions of small subvolumes to obtain a reconstruction of the complete object. Using the proposed method, the reconstruction of each subvolume is completely independent of the other subvolumes. This enables parallel computation of the complete reconstruction, resulting in a significant reduction of computation time. The results of this chapter show that the reconstruction quality of such a tiling is comparable to the standard global regularized iterative reconstruction. The reduction of computation time might enable the use of more advanced types of prior knowledge that are too computationally expensive to apply globally. Another application is to quickly estimate the parameters of a slow regularized iterative method by estimating them in only a small subvolume.

The filter-based method of Chapter 3 on which the proposed method is based relies on the shift-invariance of the projection operations. Therefore, it is only applicable to parallel-beam tomography in its current form. How to apply a similar method to other acquisition geometries is subject to further research. It may be necessary to use additional approximations to derive filter-based methods in other geometries, in which case exploiting prior knowledge may actually help to reduce artifacts caused by the additional approximations.

5

Neural network filtered backprojection

5.1 Introduction

The main problem in tomography is the reconstruction of an unknown image from its projections, acquired along a range of angles. This problem occurs in many real world applications, such as X-ray tomography in medical imaging and electron tomography in materials science. Because of its practical relevance, a large amount of research has been devoted to developing tomographic reconstruction methods (see [KS01; Nat01; Buz08] for an overview). Most common reconstruction methods can be divided into two groups: *analytical* methods and *algebraic* methods.

Analytical methods, of which filtered backprojection (FBP) is the most widely used example, are based on a continuous representation of the reconstruction problem. An analytical inverse formula of the Radon transform is discretized to obtain a reconstruction algorithm. The advantage of analytical methods is that they are usually computationally inexpensive. However, the approach is based on the assumption that the projection data is available for *all* angles, which is clearly not feasible in practice. As a result, the reconstruction quality of analytical methods tends to become unacceptable when data is only available for a small number of angles.

In several applications of tomography, practical considerations limit the number of angles for which data can be acquired. These reconstruction problems are known

---

This chapter is based on:
D. M. Pelt and K. J. Batenburg. “Fast tomographic reconstruction from limited data using artificial neural networks”. *IEEE Transactions on Image Processing* 22.12 (2013), pp. 5238–5251.

as *limited-data problems*. For example, in electron tomography, the electron beam damages the sample, imposing a strong limitation on the number of angles [MDG95]. Furthermore, in most applications, acquiring data for more projection angles requires more time. In industrial tomography, process speed considerations limit the total scan duration, making only a limited number of angles possible [Sip93]. For such problems, algorithms are needed that can create accurate reconstructions from limited data.

Algebraic reconstruction methods, such as ART and SIRT [KS01], often handle limited-data problems better than analytical methods. They are based on a discrete representation of the problem, which leads to a system of linear equations. These equations can be solved using iterative methods. Since these methods are based on a model of the data that is actually available, they can lead to more accurate reconstructions than analytical methods. The computational cost of these methods is high however, often several orders of magnitude larger than analytical methods, even when using highly optimized implementations on graphic processor units (GPUs) [XM05].

Recently, a range of algebraic methods have been developed that exploit prior knowledge about the unknown image to solve limited-data problems even more accurately. For example, total variation minimization based methods, such as FISTA [BT09a], can compute accurate reconstructions if the image has a sparse gradient [SP08]. In discrete tomography, reconstruction methods like DART [BS11] can solve limited-data problems where the original image is known to consist of only a small number of different gray levels. Although these methods produce accurate results in many cases, they have two main disadvantages: (i) they are based on algebraic methods, sharing their high computational cost; (ii) the specific prior knowledge can limit the types of images that can be reconstructed. As an example of the second point, total variation minimization methods can only accurately reconstruct objects with a sparse gradient.

In this chapter, we present a reconstruction method for limited-data problems that is specifically designed to avoid both problems. The method is computationally similar to analytical methods, ensuring a low computational cost. Furthermore, the method learns how to use problem specific knowledge to produce more accurate reconstructions than existing analytical methods. This learning is accomplished by using an *artificial neural network* (ANN). No specific prior knowledge has to be presented to the method, making it applicable to any type of image. The result is a very general method, able to produce accurate reconstructions in short time.

Artificial neural networks have been applied to tomographic reconstruction problems by several authors (see, e.g. [SH10] for an overview in the context of medical imaging). Some previous approaches have focused on directly solving a single instance of the tomographic reconstruction problem using a Hopfield neural network as an optimization tool [SHO93; Cic+95; WW97; Cie08; Cie09]. These methods compute reconstructions by minimizing the difference between the measured projection data and projections of the reconstructed object. As such, they are essentially algebraic reconstruction methods, since the objective function that is minimized is algebraic in nature. Since the neural networks have to solve a nonlinear system instead of a linear one, the reconstruction time of these methods is often even larger than the reconstruction time of linear algebraic methods.

Other previous work on using neural networks to solve tomographic problems is based on methods with a separate training phase, where the neural network is trained on a set of example images [KB95a; KB95b; Rod+01; BK06; DKM07]. In subsequent reconstruction steps, no additional training is performed. In these methods, the neural network reconstructs the entire image from the available projection data, an approach that leads to large network sizes. Because of this large size, the training phase of these methods can take a long time. Furthermore, the number of example images that the network can be trained on is typically small, limiting the reconstruction accuracy and generalizability that can be obtained (see, for example, [Rod+01]). In particular, we have not found reports on successful application of such methods to reconstruction problems involving large images (i.e. slices of 512×512 pixels or larger).

In this chapter we present a novel neural network approach to tomography, which does not have the aforementioned drawbacks. Our approach has some similarities to previous methods, such as a separate training phase, yet we use a different network model. In our model, the network reconstructs a single pixel of the reconstruction grid, using reduced projection data. This approach leads to small network sizes, which leads to fast training times, and enables us to use advanced neural network training methods. Furthermore, in our approach, each pixel of an example image can be used as an independent example during training. Therefore, we are able to use a large number of examples to train the neural network on. As a result, the trained networks yield accurate reconstructions from limited data, as well as robustness to noise.

A somewhat similar method is given in [BK06; VKB11], where the reconstruction step is implemented by using the neural network as a black box, resulting in a slow reconstruction method. In the current chapter, a different network model is chosen, such that it can be viewed as an analytical reconstruction method, having both a low computational cost and a high reconstruction accuracy. As a result, our approach can be applied to large datasets, at a computational cost that is comparable to analytical methods.

This chapter is structured as follows. In Section 5.2, we formally define the tomographic reconstruction problem and artificial neural networks. Section 5.3 introduces the new reconstruction method, which is the key contribution of this chapter. We discuss how we implemented this method in Section 5.4. In Section 5.5, we describe the experiments that we performed to compare the reconstruction time and accuracy of the new method with existing methods. The results of these experiments are given in Section 5.6, along with a discussion of these results. We conclude the chapter in Section 5.7 with a summary and some final remarks.

5.2 Notation and concepts

In this section, we will define the mathematical notation that is used in the rest of the chapter, and introduce the relevant concepts. First, we formally define the tomographic reconstruction problem, and the popular filtered backprojection algorithm. Then, we introduce artificial neural networks, the mathematical construct on which our new method is based.

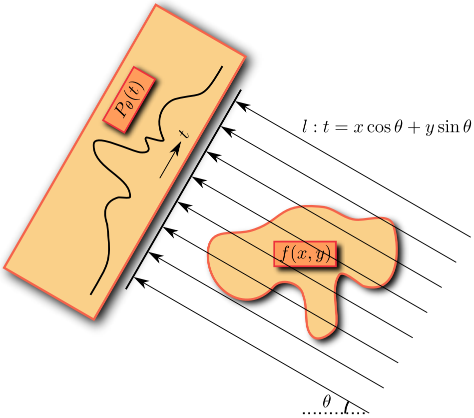


Figure 5.1: The tomography model used in this chapter. Several parallel lines, rotated by angle θ, are passing through the object f. Each line has a characteristic equation $t = x\cos\theta + y\sin\theta$, with constant t. The projection P_θ of f is given by the line integrals of f over the different parallel lines.

Problem definition

We will focus in this chapter on reconstructing two-dimensional objects from *parallel-beam* projections with a single rotation axis. The unknown object is modeled as a two-dimensional finite and integrable function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ with bounded support. We define a projection P_θ of f as the line integral of $f(x,y)$ over line l_θ:

$$P_\theta(t) = \int_{l_\theta} f(x,y)\,\mathrm{d}s \tag{5.1}$$

$$= \iint_{\mathbb{R}^2} f(x,y)\delta(x\cos\theta + y\sin\theta - t)\,\mathrm{d}x\mathrm{d}y \tag{5.2}$$

This integral transform is called the *Radon transform* of f.

Given an image $f(x,y)$, we can model the projection geometry in parallel-beam tomography as a number of parallel lines going through f, each rotated by a certain projection angle θ. A point (x,y) on one such line l_θ obeys the equation $t = x\cos\theta + y\sin\theta$. For each line l_θ, a unique constant t defines all points on that line. This model is shown graphically in Fig. 5.1. The basic tomographic problem is to reconstruct the unknown image $f(x,y)$ from the measured projections.

In practice, only discrete projection data is available, which consists of a matrix of measured values, one for each combination of N_θ projection angles $\theta \in \Theta = \{\theta_0, \theta_1, \dots, \theta_{N_\theta - 1}\}$ and N_d detectors $p \in \{0, 1, \dots, N_d - 1\}$. The position of a detector p relative to the central detector is given by τ_p:

$$\tau_p = d\left(p - \frac{N_d - 1}{2}\right), \tag{5.3}$$

where d is the width of a detector. The entire set of detector positions is given by $T = \{\tau_0, \tau_1, \ldots, \tau_{N_d-1}\}$.

The projection data is used to reconstruct f on an $N \times N$ grid of square pixels. Without loss of generality, we define that each pixel has a width and height of one, and that the center of the grid is positioned at the origin. In this case, the center of pixel (x_i, y_j) is situated in row j and column i of the pixel grid, with $i \in \{0, 1, \ldots, N-1\}$ and $j \in \{0, 1, \ldots, N-1\}$, and $x_i = y_i = i - (N-1)/2$.

Filtered back projection

One way of solving the reconstruction problem is to find a direct inverse of Eq. (5.2). To perform this inversion, we first convolve the projection data with a filter $h_\theta(t)$:

$$q_\theta(t) = \int_{-\infty}^{\infty} h_\theta(\tau) P_\theta(t-\tau)\mathrm{d}\tau \tag{5.4}$$

We can also perform this operation in the Fourier domain, where $\hat{P}$ and H denote the Fourier transforms of P and h:

$$q_\theta(t) = \int_{-\infty}^{\infty} \hat{P}_\theta(u) H_\theta(u) e^{2\pi \imath u t}\mathrm{d}u \tag{5.5}$$

By taking the formal adjoint of the Radon transform, it can be shown that if $H_\theta(u) = |u|$, we obtain a direct inverse of Eq. (5.2) [KS01]:

$$f(x,y) = \int_0^{\pi} q_\theta(x\cos\theta + y\sin\theta)\mathrm{d}\theta \tag{5.6}$$

In practice, Eq. (5.6) cannot be used directly, since $P_\theta(t)$ can only be measured for a finite set of angles Θ and a finite set of detector positions T. Therefore, we need to discretize both variables to obtain a usable reconstruction algorithm. Inserting Eq. (5.4) in Eq. (5.6) and discretizing, we obtain the filtered back projection method (FBP):

$$f(x,y) \approx FBP_{\mathbf{h}}(x,y) = \sum_{\theta_d \in \Theta} \sum_{\tau_p \in T} h(\tau_p) P_{\theta_d}(t - \tau_p) \tag{5.7}$$

where $t = x\cos\theta_d + y\sin\theta_d$. Because the projection data is discretized, interpolation is needed to obtain values at $t - \tau_p$, for $\tau_p \in T$. Linear interpolation is often adequate, since projection data is usually reasonably smooth.

The convolution operation in Eq. (5.7) can be performed in Fourier space, leading to an efficient implementation of FBP: first convolve the projection data with filter $\mathbf{h}$ in Fourier space in $\mathcal{O}(N_\theta N_d \log N_d)$ time and afterwards backproject the result to obtain the reconstruction in $\mathcal{O}(N_\theta N^2)$. Various discrete approximations of the ideal filter $H_\theta(u) = |u|$ are used in practice, such as the Ram-Lak (ramp), Shepp-Logan, and Hann filters [Far+97]. The Ram-Lak filter, obtained by setting $H_\theta(u)$ to 0 when $u > u_c$ for some u_c is often used. This filter is shown in real space in Fig. 5.2.

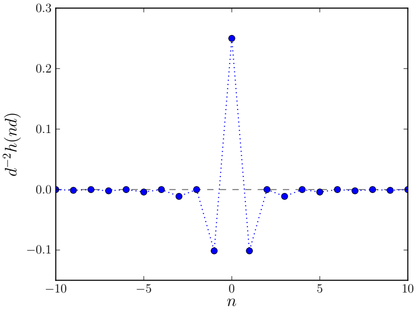


Figure 5.2: The widely used discrete Ram-Lak filter for the FBP algorithm (Eq. (5.7)). In this image, d is the distance between adjacent detector positions. This filter is an approximation of the ideal filter, obtained by taking the Fourier transform $H_\theta(u) = |u|$ of the ideal filter, and setting $H_\theta(u) = 0$ when $u > u_c$ for some u_c.

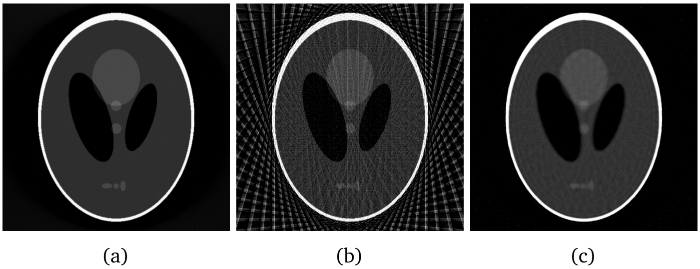


Figure 5.3: Various reconstructions of the Shepp-Logan head phantom on a 512 × 512 pixel grid. In (a) the phantom was reconstructed by FBP using 512 projection angles $\in [0, \pi)$. In (b) and (c), only 32 projection angles were used to reconstruct the phantom. FBP was used in (b), while the image (c) was obtained by using SIRT, an iterative algebraic method, with prior knowledge about the minimum and maximum possible image values.

FBP is one of the most widely used reconstruction methods in practice, because of the low computational cost compared to other methods, and good reconstruction quality if data of enough projections are available. The accuracy of the reconstructions depends on how well Eq. (5.7) approximates Eq. (5.6). If data of many projection angles are available (say, several hundreds), the approximation is often very good. When using FBP with a small number of angles, artifacts appear in the reconstructions. These artifacts can make further analysis of the reconstruction, such as segmentation, very difficult. An example of artifacts in an FBP reconstruction of limited data is shown in Fig. 5.3b. Note that a reconstruction of the same data by an algebraic method, shown in Fig. 5.3c, contains less artifacts, but takes more time to compute.

Artificial neural networks

An artificial neural network (ANN) is a computational model that processes input data using artificial neurons. The model is inspired on the workings of the human

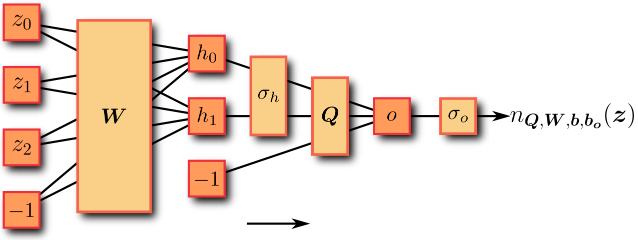


Figure 5.4: A multilayer perceptron with three input nodes z_i, two hidden nodes h_i, and one output node o. The input vector is multiplied by the weight matrix **W** to obtain hidden node inputs, and the hidden node output vector is multiplied by the weight matrix **Q** to obtain the input of the output node. Note that the biases **b** and $\mathbf{b_o}$ of Eq. (5.10) are modeled as an additional input node and hidden node of value -1. Activation functions σ_h and σ_o are applied to the hidden nodes and output node.

brain, although ANNs can also be interpreted mathematically as a class of functions. Neural networks have many uses, from simple curve fitting to complex pattern recognition [Yeg09; Hay09].

An artificial neural network can be used to model an unknown function $r : \mathbb{R}^n \to \mathbb{R}^m$. One method to accomplish this is called a *multilayer perceptron* [Hay09]. A multilayer perceptron consists of three distinct layers: the input layer, the hidden layer and the output layer. The input layer consist of n nodes, one for each input value, and the output layer has m nodes, one for each output value. The hidden layer consists of N_h hidden nodes, where N_h can be chosen freely. Generally, it is difficult to know what the optimal number of hidden nodes is for a given problem. Take too few nodes, and the network will be unable to model the unknown function. Take too many, and the resulting network will be slower and more prone to *overfitting* [TLL95]. The problem of overfitting and the way it is addressed in this chapter are explained in Section 5.4.

In a multilayer perceptron, each input node is connected to all hidden nodes, and each hidden node is connected to all output nodes. Every connection has a certain weight, and the weights can be adjusted to fit different functions r. The weights of the connections from the n input nodes to the N_h hidden nodes can be written as an $n \times N_h$ matrix **W**, where the value w_{ij} in row i and column j gives the weight of the connection between input node i to hidden node j. Similarly, the weights from hidden nodes to output nodes can be written as an $m \times N_h$ matrix **Q**. We denote a single column i of **W** as $\mathbf{w}_i$, and a single column i of **Q** as $\mathbf{q}_i$.

Scalar offsets $b \in \mathbb{R}$ are subtracted from the output of each hidden node and output node. Furthermore, nonlinear *activation functions* $\sigma_h : \mathbb{R} \to \mathbb{R}$ and $\sigma_o : \mathbb{R} \to \mathbb{R}$ are applied to the outputs of these nodes, making the entire model nonlinear in nature. In this chapter, we used the sigmoid function as activation function:

$$\sigma_h(t) = \sigma_o(t) = \frac{1}{1+e^{-t}} \tag{5.8}$$

The equation for the output of a multilayer perceptron, with a vector $\boldsymbol{z}$ as input, is given by:

$$n_{\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}}(\mathbf{z}) = \sigma_o\left(\sum_{i=0}^{N_h-1} \mathbf{q}_i g_{\mathbf{w}_i,b_i}(\mathbf{z}) - \mathbf{b_o}\right) \tag{5.9}$$

where the activation function σ_o is evaluated element-wise on its input vector, and $g_{\mathbf{w}_i,b_i}$ is the output of a hidden node:

$$g_{\mathbf{w},b}(\mathbf{z}) = \sigma_h(\mathbf{w}\cdot\mathbf{z} - b) \tag{5.10}$$

The question remains how to choose $\mathbf{Q}$, $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{b_o}$, such that $n_{\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}}(\mathbf{z}) \approx r(\mathbf{z})$. In this chapter, *supervised learning* [AB09] is used, where we assume that, although the function r is unknown, a set of T inputs $\{Z_0, Z_1, \ldots, Z_{T-1}\}$ with corresponding outputs $\{O_0, O_1, \ldots, O_{T-1}\}$ of r *are* known, where $Z_i \in \mathbb{R}^n$ and $O_i \in \mathbb{R}^m$. Learning is then defined as the minimization of the sum of squared differences between the perceptron output and the correct output:

$$e(\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}) = \sum_{i=0}^{T-1}\left(n_{\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}}(Z_i) - O_i\right)^2 \tag{5.11}$$

$$\mathbf{Q}_l, \mathbf{W}_l, \mathbf{b}_l, \mathbf{b_o}_l = \underset{\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}}{\operatorname{argmin}}\, e(\mathbf{Q},\mathbf{W},\mathbf{b},\mathbf{b_o}) \tag{5.12}$$

Because of the mathematical form of a perceptron, partial derivatives of the parameters, such as $\frac{\partial e}{\partial w_{ij}}$, can be calculated quickly and accurately by applying the chain rule. The fact that these partial derivatives are easily obtained leads to efficient applications of gradient based minimization methods to train such networks. Different methods can be used for training, each with their own advantages and disadvantages. The specific method used in this chapter is given in Section 5.4.

5.3 Neural network filtered backprojection

In this section, we present the key contribution of this chapter: the neural network filtered backprojection method (NN-FBP). We start by defining a neural network model to reconstruct a single pixel of an image. We show that this model can be viewed as a combination of FBP steps, obtaining an efficient implementation of the method. Finally, we give examples of how the new method can be used in practice.

Neural network model

To solve the basic tomographic problem using an artificial neural network, we need to define a network model: a method of converting the given projection data to input for the neural network. As explained above, we want to be able to view the chosen model as a combination of filtered backprojection steps. Therefore, it is informative to look at the equation of the FBP method:

$$FBP_{\mathbf{h}}(x,y) = \sum_{\theta_d\in\Theta}\sum_{\tau_p\in T} h(\tau_p) P_{\theta_d}(x\cos\theta_d + y\sin\theta_d - \tau_p) \tag{5.13}$$

A first observation is that Eq. (5.13) gives the value of a single point (x, y) of the FBP reconstruction. To mimic this, we choose to use a network model that reconstructs a single pixel (x_i, y_j). The neural network only has a single output node, and the output of the network is a single value in $\mathbb{R}$.

A second observation is that the FBP method is *linear shift invariant* [KS01]. Suppose we shift an object f by δx horizontally and δy vertically to obtain a shifted object f'. The original projections P_θ shift accordingly to new projections P'_θ:

$$P'_\theta(\tau) = P_\theta(\tau - (\delta x \cos\theta + \delta y \sin\theta)) \tag{5.14}$$

For the FBP reconstruction of f, denoted by $FBP^f_{\mathbf{h}}$, and the FBP reconstruction of f', denoted by $FBP^{f'}_{\mathbf{h}}$, we have:

$$FBP^{f'}_{\mathbf{h}}(x + \delta x, y + \delta y) = FBP^f_{\mathbf{h}}(x, y) \tag{5.15}$$

To mimic the linear shift invariance of FBP, we want the neural network model to treat every pixel of the reconstruction grid the same, independent of its actual position on the grid. An additional advantage of treating each pixel the same is that we can use every pixel of the grid as an independent training example during supervised learning (Eq. (5.11)). In order to accomplish this position independence, we shift the reconstructed object such that the pixel that it currently reconstructs, (x_i, y_j), is at the origin. In other words, as input for the neural network, we use projection data of the shifted object f', which can be obtained by shifting f by $-x_i$ horizontally and $-y_j$ vertically. For the projection data of the shifted object, we have (Eq. (5.14)):

$$P'_\theta(\tau_d) = P_\theta(\tau_d + x_i \cos\theta + y_j \sin\theta) \tag{5.16}$$

Now, we combine the shifted data of all projection angles by summing them element-wise:

$$P'(\tau_p) = \sum_{\theta_d \in \Theta} P'_{\theta_d}(\tau_p) \tag{5.17}$$

Finally, we reflect the shifted and summed data about the detector center:

$$z(\tau_p) = P'(-\tau_p) = \sum_{\theta_d \in \Theta} P_{\theta_d}(x_i \cos\theta_d + y_j \sin\theta_d - \tau_p) \tag{5.18}$$

The values of $z(\tau_p)$ are used as input for the neural network, as an input vector $\mathbf{z}$ with N_d elements. Note that in Eq. (5.18), only the original projection data P_{θ_d} is used. Therefore, we do not have to explicitly shift f to f' for every pixel we reconstruct, but only have to shift the original projection data by $x_i \cos\theta_d + y_j \sin\theta_d$. The transformation from projection data to network input is shown in Fig. 5.5.

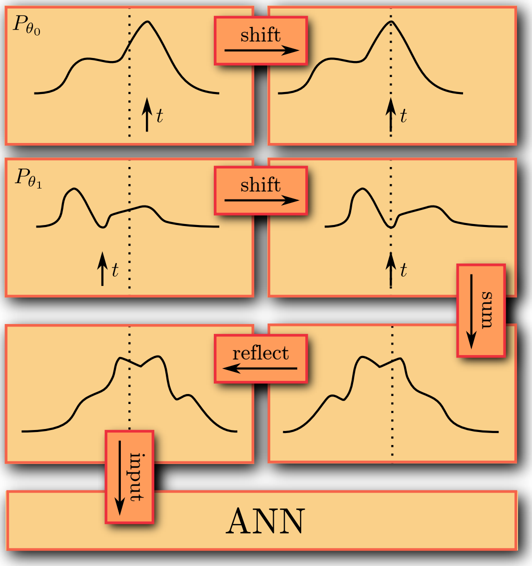


Figure 5.5: The method of transforming projection data to neural network input for pixel (x_i, y_j). For each angle θ_d, (x_i, y_j) projects onto a different point $t_d = x_i \cos\theta_d + y_j \sin\theta_d$ on the detector. We shift each projection P_{θ_d} such that the corresponding t_d is in the middle. Finally, we sum the shifted projections point by point and reflect about the center to get the network input.

Filtered back projection view

To see what the effect of the choice of network model is, we take the equation of a single hidden node $g_{\mathbf{w},b}$ (Eq. (5.10)), and insert our network model (Eq. (5.18)):

$$g_{\mathbf{w},b}(\mathbf{z}) = \sigma_h\left(\mathbf{w}\cdot\mathbf{z} - b\right) \tag{5.19}$$

$$= \sigma_h\left(\sum_k w_k \sum_{\theta_d\in\Theta} P_{\theta_d}(t-\tau_k) - b\right) \tag{5.20}$$

where $t = x_i\cos\theta_d + y_i\sin\theta_d$. Rearranging the sums and comparing with Eq. (5.7) we get:

$$g_{\mathbf{w},b}(\mathbf{z}) = \sigma_h\left(\sum_{\theta_d\in\Theta}\sum_i w_i P_{\theta_d}(t-\tau_i) - b\right) \tag{5.21}$$

$$= \sigma_h\left(FBP_{\mathbf{w}}(x_i, y_j) - b\right) \tag{5.22}$$

The entire neural network equation will now become:

$$n_{\mathbf{Q},\mathbf{W},\mathbf{b},b_o}(\mathbf{z}) = \sigma_o\left(\sum_{k=0}^{N_h-1} q_k\sigma_h\left(FBP_{\mathbf{w}_k}(x_i, y_j) - b_k\right) - b_o\right) \tag{5.23}$$

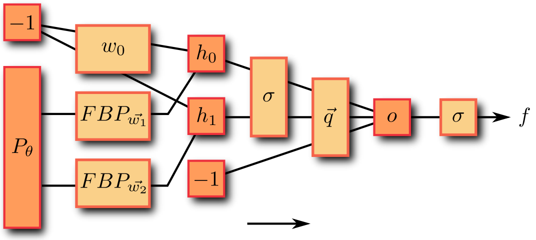


Figure 5.6: The FBP view of NN-FBP. Here, we take the projections P_θ and apply several FBP algorithms: to obtain the hidden node h_i, we apply the FBP algorithm with custom filter $\mathbf{w_i}$ and a bias. A linear combination of all hidden node images and a bias, with a sigmoid function applied to all pixels of each image, leads to a single image o. After we apply a final sigmoid function, we get an approximation of f. Note that in this case, we reconstruct the entire image f, where in the neural network view of Fig. 5.4 only a single pixel (x_i, y_j) is reconstructed.

This shows that we can view a trained network as a weighted sum of N_h FBPs with custom filters $\mathbf{w_i}$ and added biases $\mathbf{b}$. A sigmoid function is applied to the output of each FBP, and also to the final sum. The advantage of this view is that in this case, we do not have to run the network for every pixel to get the reconstruction image: we can simply apply the FBPs to obtain the entire reconstruction image in one operation.

To summarize, our new method works as follows:

Algorithm 5.1 NN-FBP reconstruction method

1. Perform N_h FBP algorithms, each with a different filter.
2. Subtract a bias from each resulting image, and apply a nonlinear activation function σ_h to each pixel of the result.
3. Multiply each resulting image with a certain weight, and add them together pixel by pixel to obtain a single image.
4. Subtract a bias from the resulting image, and apply a nonlinear activation function σ_o to each pixel to get the final reconstruction.

Note that the results of this method are identical to the results of directly applying the standard multilayer perceptron output equation (Eq. (5.9)) with Eq. (5.18) as network input. The equivalence of both methods is shown in Figs. 5.4 and 5.6.

The computational complexity of Algorithm 5.1, however, is significantly lower than direct application of Eq. (5.9). In Eq. (5.9), we need to shift, sum and reflect the input data, costing $\mathcal{O}(N_d N_\theta)$ time, for each of the N^2 pixels. Additionally, applying Eq. (5.9) takes $\mathcal{O}(N_h N)$ for every pixel, since there are $N_h N$ connections between the input layer and hidden layer. Direct application will therefore take $\mathcal{O}((N_\theta + N_h)N^3)$ time to reconstruct the entire $N \times N$ image if $N \approx N_d$.

For Algorithm 5.1, we need to perform N_h FBPs, and the computation time of step 1) is $\mathcal{O}(N_h N_\theta N^2)$. The remaining operations (adding biases, applying the activation functions and weight multiplication) each take $\mathcal{O}(N^2)$ time. Therefore, step 2) and step 3) take $\mathcal{O}(N_h N^2)$ time in total, and step 4) is $\mathcal{O}(N^2)$. We see that by exploiting the FBP view, we have reduced the reconstruction time from $\mathcal{O}((N_\theta + N_h)N^3)$ to $\mathcal{O}(N_h N_\theta N^2)$. Results from Section 5.6 will show that the method can produce accurate reconstruction even when $N_h \ll N$. Furthermore, for limited-data problems $N_\theta \ll N$, so the reduction in computation time is significant.

Training

The filters, biases and weights are trained using standard training methods from neural network theory [Hay09]. The training phase is separate from subsequent reconstruction steps: we first train the network to obtain $\mathbf{Q}_l$, $\mathbf{W}_l$, $\mathbf{b}_l$ and $\mathbf{b_o}_l$, using a set of training images. Afterwards, the trained network can be used to quickly reconstruct other images by using the method described in Algorithm 5.1, without additional training.

To perform training by supervised learning, we need a set of inputs Z with corresponding correct outputs O, where $Z_i \in \mathbb{R}^{N_d}$ is shifted and summed projection data for a single pixel and $O_i \in \mathbb{R}$ is the correct value of that pixel. This means that we need a set of projection data with corresponding correct images $f(x, y)$. This presents a problem: usually, the correct image $f(x, y)$ is unknown, since that is exactly the problem we are trying to solve. However, we can take the projection data, reconstruct it using any other method, and use the reconstruction as the correct output for learning.

This training approach can be useful in two cases:

1) N_θ-REDUCTION *use-case*: Suppose that we have a scanner that can acquire projection data along a variable number of angles. Scanning with a small number of angles is preferred, because of practical considerations. To use NN-FBP in this case, we first acquire projection data along a large number of angles for a set of representative objects. We reconstruct the images using an existing reconstruction method like FBP. Then, we train NN-FBP using these reconstructions as correct output. As input during training, we only use the projection data along a small subset of angles. After training, we can scan new objects using this small set of angles, and use NN-FBP to obtain accurate reconstructions in short time. This can be useful in many practical cases, for example to increase the time resolution of tomography of dynamic systems.

2) LIMITED-DATA *use-case*: If practical considerations limit the number of angles for which projection data can be acquired, NN-FBP can be used to lower reconstruction times. In this case, we use an advanced but slow prior-knowledge based method like TV-minimization to obtain reconstructions from the limited projection data. We then train NN-FBP using these reconstructions as correct images. In other words, we train NN-FBP to mimic a slower reconstruction method. Afterwards, we can use NN-FBP to quickly reconstruct images from similar limited-data problems.

5.4 Implementation

We will now discuss our implementation of NN-FBP that was used in the computational experiments of Section 5.5. The NN-FBP method consists of two distinct parts:

the *training* phase, and subsequent *reconstruction*. In this section, we will focus on implementation of the training, since implementing the reconstruction part is fairly straightforward: it consists of several FBPs and basic image operations. More information on implementing the FBP algorithm can be found in [BB00].

Minimization method

An important part of neural network training is the minimization of the network error (Eq. (5.12)). Several minimization algorithms are well-suited for neural network training. We used the Levenberg-Marquardt algorithm (LMA) [Mar63]. LMA is a combination of the gradient descent and Gauss-Newton algorithm, improving the stability of Gauss-Newton while retaining its fast convergence. Given a function $f_\mathbf{w}(\mathbf{x})$ with n parameters $\mathbf{w}$ and a set of m correct input-output pairs $(\mathbf{x}_i, \mathbf{y}_i)$, the method iteratively minimizes the error $e(\mathbf{w}) = \sum_i (\mathbf{y}_i - f_\mathbf{w}(\mathbf{x}_i))^2$, with the parameters at iteration $j+1$ given by $\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} + \mathbf{dw}^{(j)}$. The update vector $\mathbf{dw}^{(j)}$ is obtained by solving the LMA equation:

$$(J^T J + \lambda I)\mathbf{dw}^{(j)} = J^T \left(\mathbf{y} - f_{\mathbf{w}^{(j)}}(\mathbf{x})\right) \tag{5.24}$$

where $\lambda > 0$ and J is the $m \times n$ Jacobian matrix:

$$J = \begin{bmatrix} \frac{\partial f_\mathbf{w}(\mathbf{x_0})}{\partial w_0} & \frac{\partial f_\mathbf{w}(\mathbf{x_0})}{\partial w_1} & \cdots & \frac{\partial f_\mathbf{w}(\mathbf{x_0})}{\partial w_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_\mathbf{w}(\mathbf{x_{m-1}})}{\partial w_0} & \frac{\partial f_\mathbf{w}(\mathbf{x_{m-1}})}{\partial w_1} & \cdots & \frac{\partial f_\mathbf{w}(\mathbf{x_{m-1}})}{\partial w_{n-1}} \end{bmatrix} \tag{5.25}$$

Since $J^T J + \lambda I$ is symmetric and positive definite if $\lambda > 0$, we can use Cholesky decomposition to solve Eq. (5.24).

The parameter λ is adjusted at each iteration to ensure convergence: if $e(\mathbf{w}^{(j+1)}) > e(\mathbf{w}^{(j)})$, we increase λ to $a\lambda$ and solve Eq. (5.24) again until $e(\mathbf{w}^{(j+1)}) < e(\mathbf{w}^{(j)})$. If no such λ can be found, $\mathbf{w}^{(j)}$ is a local minimum of the error function, and LMA terminates. After an accepted update, we decrease λ to λ/a for the next iteration. In this chapter, we take $a = 10$, and start with $\lambda = 10^4$.

In the case of neural network training, the function we are minimizing is Eq. (5.11). As parameters in the LMA method we use the collection of network parameters $\mathbf{W}$, $\mathbf{Q}$, $\mathbf{b}$ and $\mathbf{b_o}$. The initial values for the parameters are calculated randomly using the Nguyen-Widrow initialization method [NW90]. In order to apply LMA, we need to calculate the Jacobian matrix J at each iteration. For neural networks, these partial derivatives can be calculated accurately and efficiently by applying the chain rule. More information on the use and implementation of LMA for neural network training can be found in [HM94].

Overfitting

A common problem that can occur when training neural networks is *overfitting*. Overfitting occurs when the neural network learns too much information about the training set. An overfitted network will be very good at solving problem instances from the training set, but relatively bad at solving instances outside the training set. In the case of NN-FBP, the method will only be able to accurately reconstruct images used in the training set, and not other, unknown, images. Of course, this is undesirable: we already know solutions to the training set problems, and we would like to be able to solve different reconstruction problems by applying NN-FBP.

The problem of overfitting is well-known in neural network theory, and several ways of preventing the problem are available. Here, we use a relatively simple, but effective method. In addition to a training set, we also use an independent *validation* set of input-output pairs during training. We then calculate the error of the validation set using Eq. (5.11) after each iteration of LMA. When this error stops improving for N_{stop} iterations, we stop the training method and return the solution with the lowest validation error. In this chapter, we use $N_{stop} = 25$. Because the training and validation set are generated independently, this prevents the network from learning too much specific information about the training set. To summarize, the training method works as follows:

Algorithm 5.2 Training method

```
1. Initialize $\mathbf{W}$, $\mathbf{Q}$, $\mathbf{b}$, and $\mathbf{b_o}$ randomly (using [NW90])
2. Iterate:
    (a) Perform LMA iteration using training set
    (b) Calculate error of validation set
    (c) If validation error has not improved for $N_{stop}$ iterations, stop iterating
3. Return $\mathbf{W}$, $\mathbf{Q}$, $\mathbf{b}$ and $\mathbf{b_o}$ which had the lowest validation error
```

Exponential binning

Neural network training is often very effective at minimizing the error of Eq. (5.11), but training can take a long time. In the case of NN-FBP, we can greatly reduce the training time by using *exponential binning*. Exponential binning was also used effectively in [BK06] to reduce the reconstruction time of the neural network.

Looking at the Ram-Lak filter of Fig. 5.2, we note that the magnitude of $h(\tau)$ is relatively large around $\tau = 0$ and drops to zero quickly for $|\tau| \rightarrow \infty$. Therefore, during reconstruction of pixel (x_i, y_j), projection data values close to $t = x_i \cos\theta + y_j \sin\theta$ are much more important than far away values. This suggests that we can reduce the number of input values by rebinning the data with a high resolution around t and a lower resolution further away. Here, we used exponential binning, where the bin width grows exponentially away from t. Formally we can define any binning by specifying the boundary points s_i and s_{i+1} of every bin: $\beta_i = (s_i, s_{i+1})$. The width of a bin is given by $d_i = s_{i+1} - s_i$. In exponential binning, we take $d_0 = 1$ and $d_i = 2^{|i|-1}$ for $i \neq 0$. A further reduction can be achieved by making the rebinning symmetric as well, by creating new bins $B_0 = \beta_0$ and $B_i = (\beta_i \cup \beta_{-i})$ for $i \neq 0$.

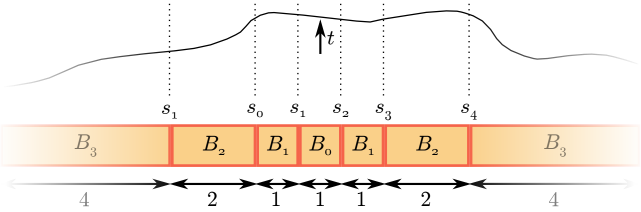


Figure 5.7: Exponential binning of the projection data, during reconstruction of a pixel (x_i, y_j), which projects onto point $t = x_i \cos\theta + y_j \sin\theta$ of the detector. Values within a bin B_i are summed to produce a single input value for the neural network. Note that the bin size increases exponentially away from t, and that the binning is symmetric, since B_i appears both to the left and to the right of t for $i \neq 0$.

To use this binning during neural network training, we apply it after the shift, sum and reflect procedure of Eq. (5.18). We sum all input values within a bin B_i to obtain the neural network input value $\bar{z}_i$:

$$\bar{z}_i = \sum_{j=s_{-i}}^{s_{-i+1}-1} z(\tau_j) + \sum_{j=s_i}^{s_{i+1}-1} z(\tau_j) \qquad \forall\, i \neq 0 \tag{5.26}$$

and $\bar{z}_0 = z(\tau_0)$. This binning procedure is shown in Fig. 5.7.

If we have N_d detectors, the output of the shift, sum and reflect procedure will have at most $2N_d$ values. We define all values outside this range to be of value 0. During binning, we only use bins that have one or both boundary points within this $2N_d$ range. Therefore, we reduce the number of input variables from $\mathcal{O}(N_d)$ to $\mathcal{O}(\log N_d)$ by using exponential binning, greatly reducing training time as well.

5.5 Experiments

In order to test the performance of the NN-FBP method, we implemented both the training and reconstruction parts using Python 2.7.3 and Numpy 1.6.3 [Oli07] built with ATLAS 3.10.0 [WP05]. We applied NN-FBP to four different problems, two for each of the two use-cases from Section 5.3. For each use-case we perform experiments on both simulation data, where the original images are known, and experimental data.

In every experiment, we are given a set of N_{im} 'correct' images, with corresponding projection data. How the correct images are obtained will be explained below for each use-case. We divide the N_{im} images into three separate groups: the *training* set, the *validation* set and the *test* set. Out of the training set, we choose N_{train} pixels to use for training, using Eqs. (5.18) and (5.26) to obtain input values for the neural network. Similarly, we take N_{val} pixels out of the validation set to use for validation, as described in Section 5.4. In this chapter, we use $N_{train} = N_{val} = 10^6$ for every experiment, unless specified otherwise.

We report results for the test set, where we use the FBP view of NN-FBP to reconstruct all test images, and report the mean absolute pixel error. The mean absolute pixel error is defined as:

$$e_p(R, O) = \frac{\langle |R - O| \rangle}{\max O - \min O} \tag{5.27}$$

where $R \in \mathbb{R}^{N \times N}$ is the reconstructed image, $O \in \mathbb{R}^{N \times N}$ the correct image, and the average is taken over all pixels that lie within the disc of radius $N/2$, centered in the image. The errors given in this chapter are the mean absolute pixel errors, averaged over all images in the test set. The results for NN-FBP are compared to results for standard FBP, with the Ram-Lak filter, and SIRT, an algebraic reconstruction method [KS01]. For both methods, we used an optimized GPU implementation from the ASTRA-toolbox [PBS11].

N_θ-reduction use-case

For the first use-case, we investigate if NN-FBP can be used to reduce the number of angles for which projection data has to be acquired. First, we reconstruct images from projection data along many angles using FBP. We then train the neural network to reconstruct these images using only a small subset of the angles. We compare the results of NN-FBP with standard FBP using the Ram-Lak filter, and with SIRT, a slower algebraic reconstruction method.

Simulation data

The simulation images used to test the performance of NN-FBP for N_θ-reduction are sampled from the threeshape family of images. Each image from the threeshape family consists of a combination of Gaussian blobs, rectangles and star-shaped objects. These components were specifically chosen to create a difficult image to reconstruct: images from the threeshape family contain both discrete and continuous areas, and both sharp edges and smooth gradients. The images are constructed as follows: starting with an image $f(x, y) = 0$, we add three Gaussian blobs, three rectangles and three star-shaped objects, each having a random shape, position, rotation and intensity. The images are then scaled, such that the darkest pixel has value 0, and the brightest has value 1. An example image of the threeshape family is shown in Fig. 5.8a.

For the training set and validation set, we generated two sets of 1000 threeshape images of 4096×4096 pixels, and calculated projection data for 4096 detector elements along 1024 equidistant angles $\in [0, \pi)$. Afterwards, we resampled the projection data to 1024 detector elements, and reconstructed on a 1024×1024 pixel grid. The test set consists of 100 images from the threeshape family. We test the network by training it to use only $N_\theta = 8, 16, 32, 64$ equidistant angles.

Experimental data

The dataset we used for experimental data stems from a small fatigue test sample made from Ti alloy VST 55531. The sample has been scanned in a parallel, monochromatic (52 keV) synchrotron X-ray beam at beamline ID11 of the European Synchrotron Radiation Facility (ESRF). The sample to detector distance was set to 40 mm and 1500 projections were acquired on a high resolution detector system. 2×2 binning resulted in projections with 1024^2 pixels and an effective pixel size of 0.56 microns. For training, validation, and testing, data from three different time steps were used, with 438 slices each.

Limited-data use-case

For the second use-case, where only a small number of projections can be acquired, we performed experiments to investigate if NN-FBP can be used to mimic advanced reconstruction algorithms that require a long computation time. We first reconstruct images using FISTA [BT09a], a TV-minimization algorithm. We train NN-FBP to approximate the result of FISTA. Afterwards, we reconstruct the test set using FISTA and NN-FBP, and report the mean absolute pixel error between the FISTA and NN-FBP reconstructions. To see the improvement of NN-FBP over standard FBP, we also report the mean absolute pixel error between FISTA and FBP.

Simulation data

For the simulation data, we sampled images from a specific family of images. Since we are investigating whether NN-FBP can mimic a TV-minimization method, images from this family should be well-suited for TV-minimization, and have a sparse gradient. Note that the threeshape family of Section 5.5 is not suitable, as the Gaussian blobs do not have a sparse gradient. Instead, we chose the 7ellipses family of images, where each image consists of 7 overlapping ellipses of random shape, position, rotation and intensity. We use 1024 × 1024 pixel images, randomly sampled from the 7ellipses family. The training, validation, and test sets consist of 100 images each. We calculate projection data for 1024 detector elements along $N_\theta = 8, 16, 24, 32$ equidistant angles. For reconstruction, we resampled the projection data to 256 detector elements, and obtained reconstructions using FISTA on a 256 × 256 pixel grid. These reconstructions were used to train the NN-FBP method, and to report errors on. An example image of the 7ellipses family is shown in Fig. 5.8c

Experimental data

Here, we use a set of experimental μCT data. These datasets were acquired by scanning raw diamonds in a Scanco 40 μCT scanner. The acquired cone-beam projection data was rebinned to a parallel beam geometry. The resulting projection data consists of 1024 detector elements along 500 projection angles, acquired for a number of two-dimensional slices through the diamonds. In total, three datasets of different diamonds were used: one for training, one for validation and one for testing. The number of slices for each dataset are 629, 358, and 375, respectively. An example of a single slice is shown in Fig. 5.8d. To test the limited-data case, we took 8, 16, 32, and 64 angles out of the available angles, resampled the projection data to 256 detector elements, and created reconstructions using FISTA on a 256 × 256 pixel grid. All calculations were performed using these reconstructions, thereby training NN-FBP to approximate the FISTA reconstructions.

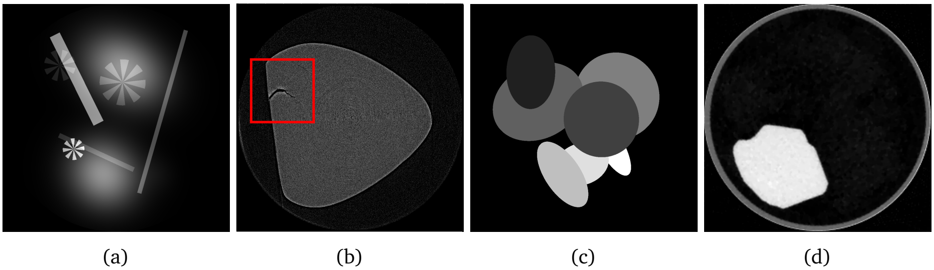

(a) (b) (c) (d)

Figure 5.8: Example images of the four experiments that we performed in this chapter. The left two images were used for the N_θ-reduction use-case, and the other two for the limited-data use-case, as explained in Section 5.5. The images of (a) and (c) are computer-generated simulation images, and the images of (b) and (c) are reconstructions of experimental CT data. The area indicated in (b) is the area of which results are shown in Figs. 5.10e to 5.10h.

5.6 Results and discussion

The mean absolute error for each use-case, averaged over the entire test set, is given in Fig. 5.9. The figure shows that for all experiments and number of hidden nodes, NN-FBP produces images with lower mean absolute error than those produced by FBP *and* SIRT. An important observation is that the improvement of NN-FBP over standard FBP is significant. Furthermore, NN-FBP with one hidden node is able to produce images with significantly lower mean absolute error compared to FBP, even though their computation complexities are identical. Although FBP with the Shepp-Logan or Hann filter performed better than FBP with the Ram-Lak filter, the NN-FBP method produced significantly more accurate reconstructions than both.

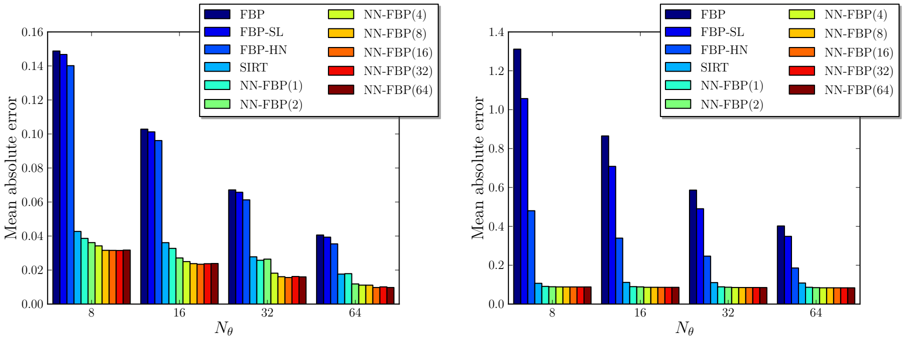
The dependence of the accuracy of NN-FBP on the number of hidden nodes N_h can be explained as follows: if not enough hidden nodes are used, the network is not able to capture all useful information during training, and the reconstruction quality suffers. If too many hidden nodes are used, the network is still able to capture all information, and reconstruction quality is still good. Since there are more weights to train, however, networks with too many hidden nodes are more difficult and time-consuming to train. With more weights, the risk of ending up in local minima of the objective function is higher, which explains why the mean absolute error sometimes increases slightly when more hidden nodes are used.

In the remainder of this section, we give detailed results for each of the use-cases, and give results of other experiments investigating the properties of NN-FBP.

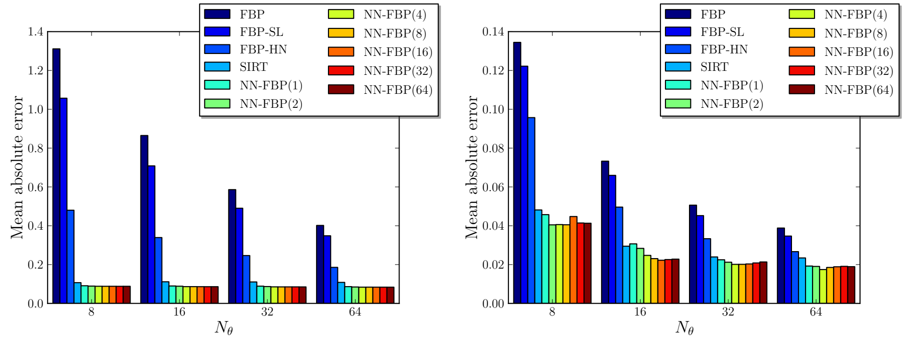
N_θ-reduction use-case

Simulations

The results for the N_θ-reduction case with simulation data is shown in Table 5.1. The results show that, for all numbers of angles, NN-FBP produces more accurate reconstructions than both FBP and SIRT. The reconstruction time of NN-FBP is close



(a) N_θ-REDUCTION, simulation data (b) N_θ-REDUCTION, experimental data



(c) LIMITED-DATA, simulation data (d) LIMITED-DATA, experimental data

Figure 5.9: The mean absolute error, averaged over the entire test set, for each use-case. Given are results for FBP with the Ram-Lak filter (FBP), FBP with the Shepp-Logan filter (FBP-SL), FBP with the Hann filter (FBP-HN), SIRT, and NN-FBP, where the number of hidden nodes is given between parentheses.

		$N_\theta = 8$			$N_\theta = 16$			$N_\theta = 32$			$N_\theta = 64$		
	N_h	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$
FBP		0.149	0.02		0.103	0.02		0.067	0.02		0.041	0.03	
SIRT		0.043	29.64		0.036	35.42		0.028	48.37		0.018	70.72	
NN-FBP	1	0.039	0.04	2330	0.033	0.04	2362	0.026	0.04	2428	0.018	0.04	2559
NN-FBP	2	0.036	0.06	2499	0.027	0.06	2557	0.026	0.07	2550	0.012	0.08	2815
NN-FBP	4	0.034	0.12	2532	0.025	0.12	2669	0.018	0.13	2630	0.011	0.15	2905
NN-FBP	8	0.032	0.23	2928	0.024	0.23	2873	0.016	0.25	3147	0.011	0.29	2912
NN-FBP	16	0.032	0.44	3092	0.023	0.45	3552	0.016	0.49	3940	0.010	0.56	3681
NN-FBP	32	0.032	0.88	4094	0.024	0.96	4527	0.016	1.04	5160	0.010	1.21	5801
NN-FBP	64	0.032	1.92	7101	0.024	1.79	9027	0.016	2.01	12273	0.010	2.28	11883

Table 5.1: Results for the N_θ-reduction use-case, simulation data. $\langle e_p \rangle$, T_r, and T_t denote mean absolute error, reconstruction time, and training time, respectively.

		$N_\theta = 8$			$N_\theta = 16$			$N_\theta = 32$			$N_\theta = 64$		
	N_h	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$
FBP		1.311	0.03		0.865	0.03		0.586	0.03		0.402	0.04	
SIRT		0.107	42.97		0.111	52.59		0.111	73.15		0.108	108.17	
NN-FBP	1	0.091	0.05	750	0.090	0.05	801	0.089	0.05	777	0.086	0.06	881
NN-FBP	2	0.089	0.09	818	0.089	0.09	770	0.087	0.10	795	0.085	0.11	927
NN-FBP	4	0.088	0.16	845	0.086	0.17	855	0.085	0.18	932	0.084	0.21	1076
NN-FBP	8	0.088	0.32	912	0.086	0.33	979	0.085	0.35	1003	0.084	0.40	1148
NN-FBP	16	0.088	0.62	1181	0.086	0.64	1392	0.085	0.69	1457	0.084	0.79	1923
NN-FBP	32	0.088	1.25	2343	0.086	1.27	2551	0.085	1.39	3221	0.084	1.56	3339
NN-FBP	64	0.088	2.60	4538	0.086	2.71	6229	0.085	2.98	8265	0.084	3.11	5788

Table 5.2: Results for the N_θ-reduction use-case, experimental data. See Table 5.1 for more information.

to the reconstruction time of FBP multiplied with a factor of N_h. For example, using NN-FBP with 8 hidden nodes, the mean absolute error is, on average, roughly 75% lower than FBP and 35% lower than SIRT. The reconstruction time for that case is 11.5 times larger than that of FBP, but only 0.6% of that of SIRT. An example image with reconstructions for $N_\theta = 32$ and $N_h = 8$ is shown in Figs. 5.10a to 5.10d, where we see that the NN-FBP reconstruction is sharper than that of SIRT, and has less streak artifacts than the FBP reconstruction.

Experimental data

For the N_θ-reduction case and experimental data, results are given in Table 5.2. Again, NN-FBP produces more accurate results than both FBP and SIRT, although the differences are smaller than for the simulation data. Images of the reconstructions close to the forming crack, given in Figs. 5.10e to 5.10h, show, however, that the reconstruction of NN-FBP is visually much clearer than the FBP and SIRT reconstructions. The FBP reconstruction suffers from the combined effect of limited data and noise, resulting in a very noisy reconstruction.

Limited-data use-case

Simulations

Results for the limited-data use-case and simulation data are given in Table 5.3. Here, the reported mean absolute errors are calculated with respect to the FISTA reconstructions. Note that the images are smaller than the ones used in Section 5.6. The reconstruction time of NN-FBP is only a fraction of the reconstruction time of FISTA, while reconstructions created by NN-FBP have a relatively low mean absolute error compared to the FISTA reconstructions. Example reconstruction are given in Figs. 5.10i to 5.10l. Compared to FBP and SIRT, the NN-FBP method is able to approximate FISTA reconstruction more accurately, although NN-FBP is not able to mimic FISTA exactly.

		$N_\theta = 8$			$N_\theta = 16$			$N_\theta = 32$			$N_\theta = 64$		
	N_h	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$
FISTA		0.000	23.5		0.000	38.0		0.000	51.2		0.000	58.7	
FBP		0.180	0.00		0.111	0.00		0.066	0.00		0.039	0.00	
SIRT		0.068	1.60		0.048	1.66		0.034	1.85		0.024	2.19	
NN-FBP	1	0.048	0.00	46	0.040	0.00	40	0.030	0.00	44	0.021	0.00	46
NN-FBP	2	0.040	0.02	108	0.031	0.01	122	0.023	0.01	192	0.016	0.01	240
NN-FBP	4	0.038	0.01	246	0.028	0.01	295	0.021	0.01	258	0.015	0.02	165
NN-FBP	8	0.038	0.02	343	0.027	0.02	500	0.019	0.03	436	0.014	0.04	436
NN-FBP	16	0.037	0.04	829	0.027	0.04	643	0.019	0.05	847	0.014	0.07	673
NN-FBP	32	0.037	0.08	1560	0.027	0.08	1557	0.019	0.11	1579	0.015	0.14	1627
NN-FBP	64	0.036	0.16	4710	0.027	0.17	4113	0.019	0.22	3904	0.015	0.29	4375

Table 5.3: Results for the limited-data use-case, simulation data. See Table 5.1 for more information.

		$N_\theta = 8$			$N_\theta = 16$			$N_\theta = 32$			$N_\theta = 64$		
	N_h	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$	$\langle e_p \rangle$	$T_r(s)$	$T_t(s)$
FISTA		0.000	24.1		0.000	30.6		0.000	41.1		0.000	54.6	
FBP		0.134	0.00		0.073	0.00		0.051	0.00		0.039	0.00	
SIRT		0.048	1.60		0.029	1.66		0.024	1.85		0.023	2.19	
NN-FBP	1	0.046	0.00	156	0.031	0.00	146	0.022	0.00	161	0.019	0.01	183
NN-FBP	2	0.040	0.01	187	0.028	0.01	211	0.021	0.01	199	0.019	0.01	208
NN-FBP	4	0.041	0.01	377	0.025	0.01	371	0.020	0.01	261	0.017	0.02	373
NN-FBP	8	0.041	0.02	585	0.023	0.02	327	0.020	0.03	779	0.019	0.04	583
NN-FBP	16	0.045	0.04	1247	0.022	0.04	1166	0.020	0.05	1058	0.019	0.07	896
NN-FBP	32	0.041	0.08	2835	0.023	0.08	3586	0.021	0.11	2605	0.019	0.14	3058
NN-FBP	64	0.041	0.16	13190	0.023	0.17	3901	0.021	0.22	5226	0.019	0.29	4831

Table 5.4: Results for the limited-data use-case, experimental data. See Table 5.1 for more information.

Experimental data

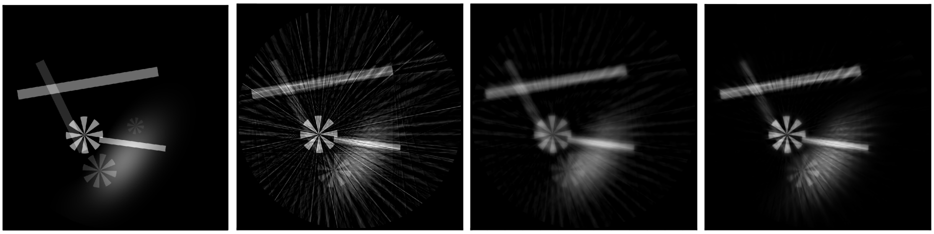
Results for the limited-data use-case and experimental data, given in Table 5.4, show similar results, where the NN-FBP reconstructions approximate the FISTA reconstructions more accurately than both FBP and SIRT. Again, it takes significantly more time to reconstruct the images using FISTA than to reconstruct them using NN-FBP. Reconstructions of a single slice of the data are given in Figs. 5.10m to 5.10p.

Other experiments

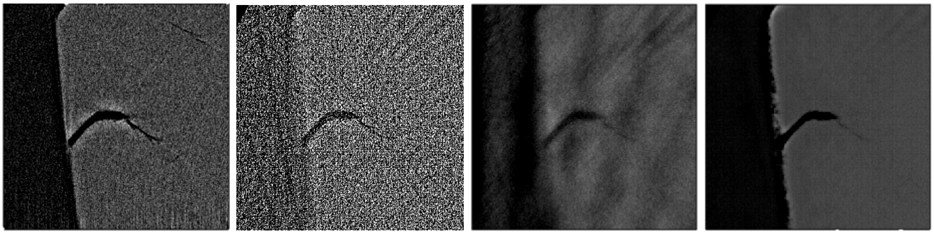
We will now discuss other experiments we performed to determine the properties of the NN-FBP reconstruction method.

Size of the training and validation set

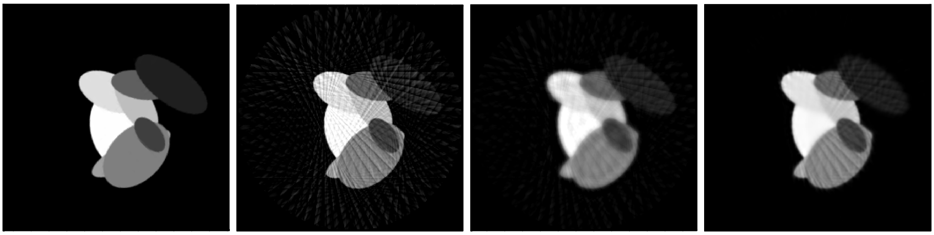
To investigate the training and reconstruction properties of the NN-FBP method, we took the N_θ-reduction use-case with simulation data over 16 angles, and the NN-FBP method with 8 hidden nodes. We trained the method 10 times, starting each time with random weights, for different sizes of the training and validation set, and calculated



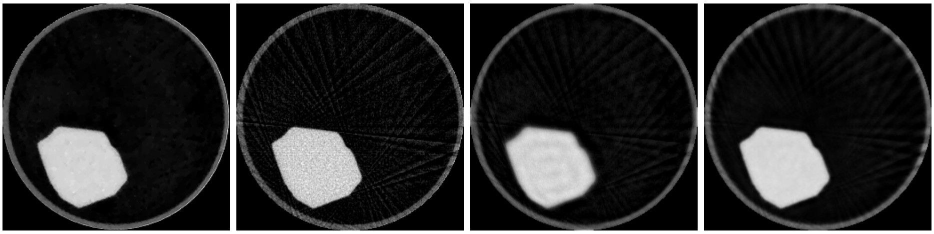
(a) Original (b) FBP (c) SIRT (d) NN-FBP



(e) Original (f) FBP (g) SIRT (h) NN-FBP



(i) Original (j) FBP (k) SIRT (l) NN-FBP



(m) Original (n) FBP (o) SIRT (p) NN-FBP

Figure 5.10: Reconstructions of the objects in the left column, obtained from projection data over 32 angles by FBP, SIRT, and NN-FBP with 8 hidden nodes. In the bottom two rows, the original object was obtained by applying the FISTA algorithm on the full set of 32 available projections.

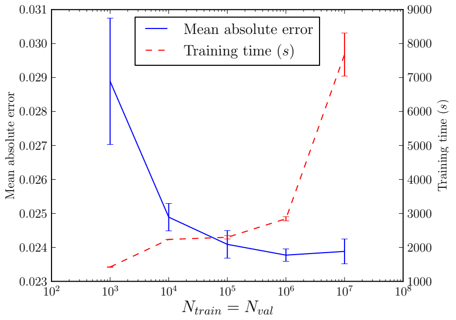


Figure 5.11: The average mean absolute error and training time for 10 runs of the N_θ-reduction use-case, simulation data, with $N_\theta = 16$ and $N_h = 8$. The error bars indicate standard deviation.

the resulting mean absolute error with the test set, and measured the training time. Results are given in Fig. 5.11.

The results show that the mean absolute error of the resulting trained network decreases with increasing training set and validation set size. After a certain size, however, increasing the size further does not seem to lower the error significantly. The time it takes to train NN-FBP becomes larger with increasing set size. Figure 5.11 also shows that for sufficient set sizes, the standard deviation of the mean absolute error is low. This is important for practical applications, since it shows that one has to train NN-FBP only once, without risk of obtaining a badly trained network.

Noise in the projection data

To investigate the effect of noise in the projection data on NN-FBP, we added different levels of Poisson noise to the simulation data of the N_θ-reduction use-case. FBP reconstructions of the noisy projection data with 1024 projection angles were used as training examples for training the NN-FBP method. After training NN-FBP to reconstruct using only 32 projection angles of the noisy data, we reconstructed a single image of the test set, and calculated the mean absolute error of the reconstruction, compared to the noiseless phantom image (Fig. 5.10a). Results are given in Fig. 5.12. The reconstructions obtained by NN-FBP are more accurate than both FBP and SIRT for all noise levels, with the mean absolute error being much lower than FBP. The artifacts in the FBP reconstructions would make further analysis of the object difficult, especially at high noise levels.

To investigate the effect of noise on the training phase of NN-FBP, we trained the NN-FBP method 10 times on a single data set, each time with independently generated noise applied. In every run, the network was trained on 10^6 pixels from a training and validation set of 100 images of the threeshape family, generated on a 1024 × 1024 pixel grid, with projection data of 32 angles, rebinned to 256 detectors. FBP reconstructions of the noisy projection data with 1024 projection angles were used as training examples. For a test set of 100 images similar to the training set, we

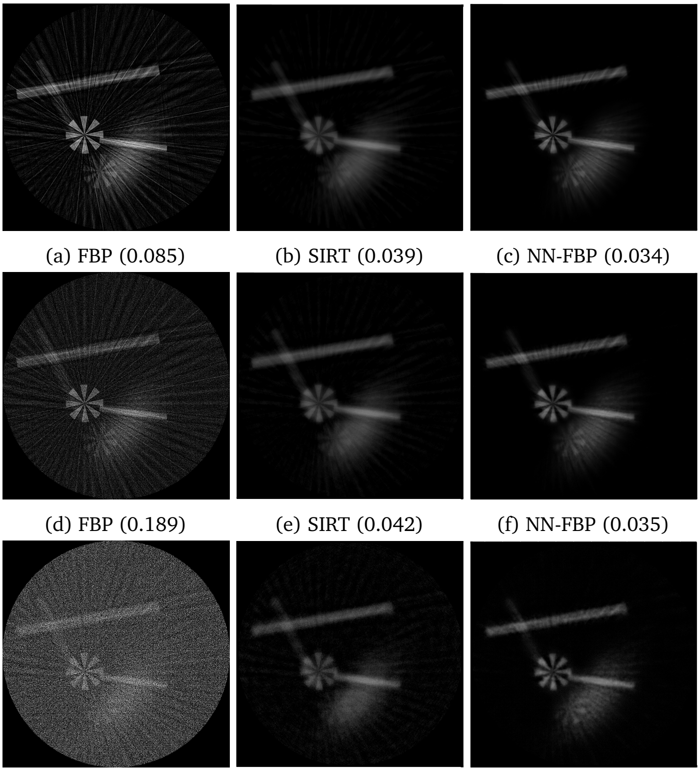

(a) FBP (0.085) (b) SIRT (0.039) (c) NN-FBP (0.034)
(d) FBP (0.189) (e) SIRT (0.042) (f) NN-FBP (0.035)
(g) FBP (0.558) (h) SIRT (0.058) (i) NN-FBP (0.043)

Figure 5.12: Reconstructions of the object from Fig. 5.10a, obtained from projection data over 32 angles with Poisson noise by FBP, SIRT, and NN-FBP with 8 hidden nodes. Each row has an increasing amount of added noise. The mean absolute error of the reconstructions, compared to Fig. 5.10a, is given between parentheses. The errors of FBP with the Shepp-Logan filter are 0.078, 0.157, and 0.452, for increasing amount of added noise. For FBP with the Hann filter, errors are 0.062, 0.090, and 0.216, respectively.

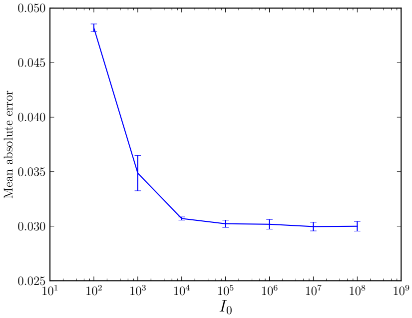


Figure 5.13: The average mean absolute error for 10 runs of the N_θ-reduction use-case, simulation data (256 × 256 pixels), with $N_\theta = 32$, $N_h = 8$, and Poisson noise. The Poisson noise is generated independently for each of the 10 runs. Lower values of I_0 correspond to larger amounts of noise. Error bars indicate standard deviation.

report the average mean absolute error of the noiseless phantom with the NN-FBP reconstructions, which calculated using noisy projections.

Results are given in Fig. 5.13. These results show that the mean absolute error decreases smoothly with decreasing noise levels. Furthermore, the standard deviation of the mean absolute error is relatively small compared to the error itself, for all noise levels. This indicates that noise in the projection data does not have a large impact on the ability of NN-FBP to find filters that minimize the training error. One reason for this robustness could be that we are able to use a large number of training examples, thereby reducing the influence of the noise by averaging its effect on each example.

Hidden node output

To gain a better insight in how NN-FBP is able to produce accurate reconstructions, we can look at the output of the hidden nodes of the network. Since the neural network of NN-FBP reconstructs a single pixel, we can view the output value of a single hidden node as a pixel of an image. In other words, we can look at the FBP reconstructions of each hidden node of Eq. (5.23). To obtain the final output of NN-FBP, these individual reconstructions are added together with an additional constant offset, and the sigmoid function is applied to each pixel value.

Figure 5.14 shows four of the eight hidden node output images, resulting from a reconstruction of Fig. 5.10i with data for 32 projection angles and NN-FBP with 8 hidden nodes. The results show that each hidden node reconstructs a different feature of the final reconstruction: some focus on the broad shape of the object, while others focus on the edges. Furthermore, the relative contrast of the different ellipses in the reconstructed object differs for each hidden node output image. These results, in addition to the other results in this section, show that there is something to gain by using multiple nonlinear FBPs to reconstruct an object, compared to using a single standard FBP.

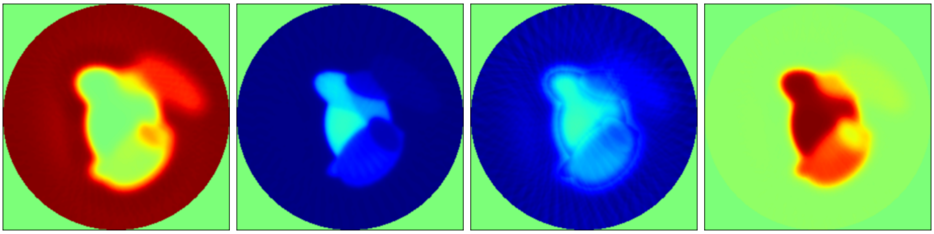


Figure 5.14: Hidden node output images of a reconstruction of the object from Fig. 5.10i, obtained from projection data over 32 angles, with 8 hidden nodes. Blue, green, and red indicate negative, zero, and positive values, respectively.

Exponential binning

To test the influence of exponential binning on both the reconstruction quality and training time of the NN-FBP method, we trained NN-FBP both with and without exponential binning. Both times, the network was trained on 10^6 pixels from a training and validation set of 100 images of the threeshape family, generated on a 1024×1024 pixel grid, with projection data of 32 angles, rebinned to 256 detectors. After training, both networks were used to reconstruct a test set of 100 images similar to the training and validation images.

With exponential binning, training the network took 673 seconds, and the resulting mean absolute error with the test set was equal to 0.0246. Without exponential binning, the mean absolute error with the test set was 0.0239, which is 3% lower. The time to train the network, however, increased to 55178 seconds, 82 times longer than with exponential binning. These results show that, although exponential binning can slightly impact the reconstruction quality of NN-FBP, it greatly reduces the time it takes to train the method.

5.7 Conclusion

In this chapter, we presented a new reconstruction method, the neural network filtered backprojection method (NN-FBP), for limited-data 2D parallel-beam tomography problems. The method is based on artificial neural networks, which allows it to learn problem specific knowledge to improve its reconstruction quality. Furthermore, we showed that NN-FBP can be viewed as a combination of several standard FBP operations, each with a custom filter. This property ensures that the computation complexity of NN-FBP is low compared to algebraic reconstruction methods.

In order to train the NN-FBP method, a set of reconstructions with corresponding projection data is needed. Although this requirement presents a problem, it can be satisfied in several practical applications. Here, we focused on two such applications. In one, we first acquire projection data over a large number of angles, and use reconstructions obtained by standard FBP to train NN-FBP on, while using limited data of only a small subset of angles. Afterwards, NN-FBP can reconstruct limited data of similar objects accurately. In the second use-case, we assume that we are not able to acquire data over a large number of angles, but are given limited-data of only a small number of angles. In this case, we can use NN-FBP to imitate a much slower prior-knowledge based reconstruction method, such as TV-minimization methods.

Results for simulation data and experimental data of both use-cases show that NN-FBP is able to produce significantly more accurate reconstructions than standard FBP. The reconstruction time of NN-FBP is slightly higher than the reconstruction time of FBP multiplied by the number of hidden nodes. The results show that even for a low number of hidden nodes, NN-FBP is able to outperform FBP. Interestingly, NN-FBP is also able to produce more accurate reconstructions than SIRT, a much slower iterative algebraic method. Additional experiments show that the method is more robust than FBP when faced with noisy projection data.

The current study focused on two-dimensional parallel-beam tomographic problems, but a similar method can in theory be applied to other tomographic problems. In these cases, the method will be related to other filter-based analytical reconstruction methods. For example, in three-dimensional cone-beam tomography problems, we can design a neural network that can be viewed as a combination of Feldkamp-David-Kress (FDK) operations [FDK84] with custom filters. Similarly, the current method can also be applied to fan-beam problems, with the fan-beam variant of FBP [KS01]. The reconstruction quality of these new methods remains subject of further research.

NN-FBP can also be used to combine the reconstruction of an object with subsequent analysis of the reconstruction. This can be achieved by training NN-FBP using analyzed images of the correct reconstructions as training images. For example, we can train NN-FBP on segmented images of the training data, thereby training it to perform both the reconstruction *and* segmentation in a single step. Other analyses, such as highlighting areas of interest, are also possible. Which type of object analyses can be accurately performed by NN-FBP remains subject of further research.

Since NN-FBP consists only of FBP operations and image addition and multiplication, implementation of the method in current applications is straightforward. Many hardware CT-scanners currently use FBP as their main reconstruction method, which would make replacement with NN-FBP easy, provided that the user is able to specify custom filters. If a heavily optimized version of FBP is available, NN-FBP will be able to use the same optimizations to reduce execution time. The results from this chapter show that NN-FBP can be a significant improvement over FBP for practical applications.

6

Application of NN-FBP in Electron Tomography

6.1 Introduction

Gold nanoparticles (NPs) have truly unique electronic, optical as well as catalytic properties, rendering them ideal for numerous applications in fields as diverse as photovoltaics, optoelectronics and biomedicine [ZCG09; ZO11; RM05; CL14]. Furthermore, gold NPs can be prepared with almost any desired shape. Crucial to their application, however, is their exact structure, and specifically their anisotropy as well as the surface facets they expose. Currently, it is empirically understood how particle size and shape may be controlled during synthesis [Pér+05; Grz+08; Sán+06; THY08]. Although transmission electron microscopy (TEM) has become a routine tool to investigate e.g. particle size, (atomic) structure and shape, increasingly advanced TEM is required for a more in-depth characterization. For example, the surface facets of Au nanorods have a major influence on crucial effects such as reactivity and ligand adsorption and there has been controversy regarding facet indexing [Pec+08; Car+10; Kat+10]. Indeed, TEM images are only two-dimensional (2D) projections of three-dimensional (3D) objects. To overcome this problem, 3D electron microscopy, or "electron tomography" was developed [Kos+97; Fra92]. In 2003, Paul Midgley and co-workers demonstrated the potential of the technique in materials science based on high angle annular dark field scanning transmission electron (HAADF-STEM) microscopy [MW03; MD09]. Since

---

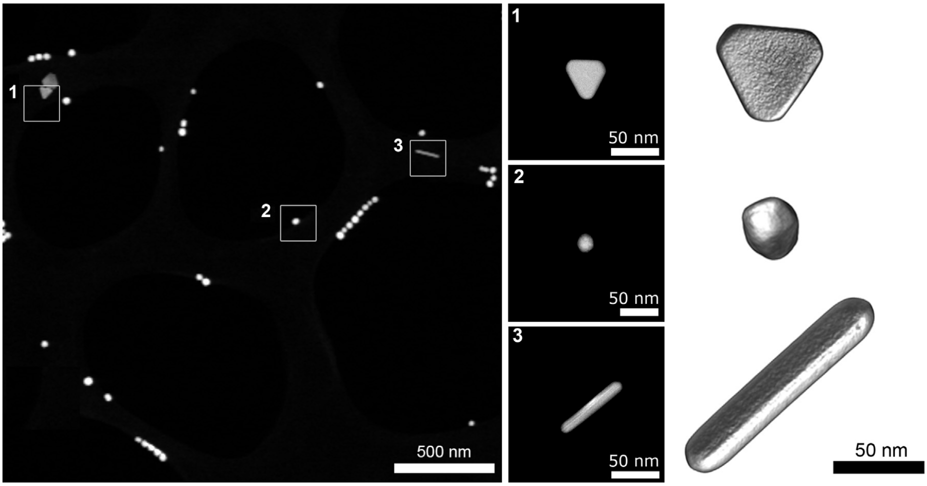
This chapter is based on:
E. Bladt, D. M. Pelt, S. Bals, and K. J. Batenburg. "Electron tomography based on highly limited data using a neural network reconstruction technique". *Ultramicroscopy* 158 (2015), pp. 81–88.

then, different electron microscopy modes have been combined successfully with tomography, leading to a broad variety of 3D structural and compositional information at the nanoscale [Wol+10; Wey+06; Bar+08; MB12; Gor+11; Kos+00]. Very often, electron tomography is used to determine the size and shape of the particles and nowadays, 3D reconstructions can even be obtained with a resolution at the atomic level [Gor+12a; Van+11]. Although these investigations provide very precise information on the NP morphology, both the acquisition of tilt series as well as the 3D reconstruction is very time consuming and it is consequently not straightforward to acquire results in 3D that are statistically relevant, which is a major drawback e.g. when using electron tomography to optimize the synthesis of NPs. This problem will be even more essential for anisotropic NPs that are currently receiving a lot of attention because of the increased flexibility they provide to tune the final (optical) properties [GS07; NPK10; GGL12]. Since the optimization of the production of NPs with a specific shape would largely benefit from statistical 3D results with a nanometer resolution, one of the emerging challenges in the field of electron tomography is to increase the throughput of 3D reconstructions of NPs. At the same time, the quality of the reconstructions should be maintained and should enable one to obtain reliable and quantitative results concerning parameters such as particle size and surface morphology.

In this chapter, we will determine the 3D shape and size of a large set of anisotropic Au NPs. We will make effective use of a new approach for electron tomographic reconstructions that is based on artificial neural networks. The neural network filtered backprojection method (NN-FBP) is a recently developed reconstruction technique that has been applied successfully to X-ray tomography (see Chapter 5); however the implementation for electron tomography is completely new. The method that we propose will enable us to reduce the number of necessary projection images for a 3D reconstruction by a factor of 5 or more. In this manner, the acquisition time and time that is necessary for a 3D reconstruction is significantly reduced, enabling 3D results that are of statistical relevance.

6.2 Neural network filtered backprojection method

The sample that was investigated contains Au NPs yielding different morphologies: nanorods, nanotriangles, nanoprisms and nanospheres. An HAADF-STEM overview image of the sample is provided in Fig. 6.1.a. Although this image only corresponds to a 2D projection of a set of 3D objects, it is already clear that different morphologies occur. In conventional electron tomography, a large set of 2D projection images is acquired from the same region of interest over a large tilt range with a tilt increment of typically 1° or 2°. As all the investigated nanoparticles have a thickness below 100 nm, the projection requirement for tomography is satisfied [MW03; Erc+06]. Once this so-called “tilt series” is aligned, the images serve as an input for a mathematical algorithm that enables one to reconstruct the original 3D structure. Very often, the 3D reconstruction is performed using the “Weighted Backprojection” algorithm (also known as Filtered Backprojection) or using the “Simultaneously Iterative Reconstruction Technique” (SIRT). The outcome of this procedure for the different NPs in Fig. 6.1.a is



(a) (b)

Figure 6.1: (a) The HAADF-STEM overview image shows the presence of several morphologies in the sample, with indication of (1) a nanotriangle, (2) a nanosphere and (3) a nanorod. (b) 3D volume renderings of the corresponding nanoparticles are presented.

visualized in Fig. 6.1.b. The reconstructions are calculated using the SIRT algorithm and are based on a series of 151 images, acquired over a tilt range of ± 75°. Since the quality of 3D reconstructions based on the conventional approach is predominantly determined by the number of projection images [CDK70; Gil72; MVB11], these experiments are very time-consuming and require sufficient measurement time at the TEM.

The key to increasing the image quality if only a small number of 2D projections are available, is the effective use of prior knowledge in the reconstruction. By exploiting rather generic features of the particles, without assuming a specific shape or morphology, this additional knowledge is used to compute a particle shape that better approximates the true morphology. Various algorithms involving prior knowledge are currently in use in electron tomography (e.g. the DART algorithm for discrete tomography [BS11] and multiple methods for Total Variation Minimization [Gor+12b]), where the particular prior knowledge is encoded by the user and various parameters have to be set. These prior-knowledge based methods are typically very time-consuming, which limits the throughput of 3D reconstructions that can be achieved by using them for reconstruction. Furthermore, implementing these methods can be difficult and time-consuming as well, since they rely on advanced mathematics. In this chapter, we propose an alternative approach called Neural Network Filtered Backprojection (NN-FBP), described in Chapter 5 of this thesis, which can effectively exploit sample characteristics to improve reconstruction quality, while still being highly computationally efficient. Here, we apply this new technique for the first time to electron tomography data. The application of NN-FBP to electron tomography consists of two phases: (i) a learning phase, in which full tilt series and their corresponding reconstructions are used to calibrate the

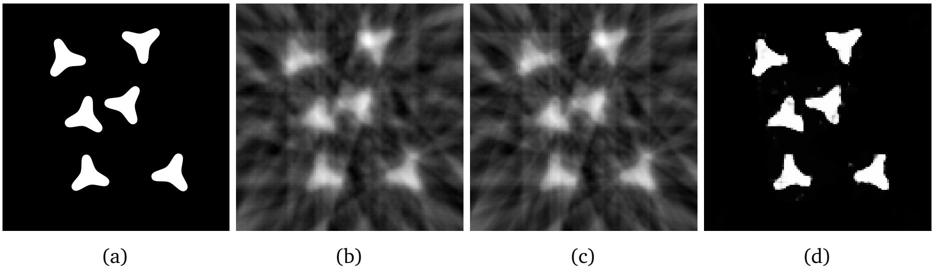


Figure 6.2: Three reconstructions of a phantom image from 10 projections: (a) the phantom image, (b) WBP with a single filter, (c) a linear combination of two WBP reconstructions, and (d) a combination of two WBP reconstructions with a pixel-wise nonlinear scaling operation. In each reconstruction, the weights and filters are chosen such that the mean squared error with the phantom image is minimized.

reconstruction algorithm and (ii) a reconstruction phase, in which large batches of limited tilt series (i.e. using fewer projections) are rapidly reconstructed. In the next subsections, we will first briefly explain how the reconstructions are formed in the reconstruction phase, followed by an overview of how the calibration is performed in the learning phase.

Reconstruction phase

Reconstructions obtained by standard Weighted Backprojection are commonly plagued by a range of reconstruction artifacts when reconstructing from a limited tilt range and few projection angles. Streaks can be observed due to the limited number of projections, and the limited angular range leads to elongation and blurring in the Z-direction. In Chapter 5, it was found that strong improvements on the reconstruction quality from limited data can be obtained by combining a small number (e.g. 2 or 4) of WBP reconstructions, each obtained using a different filter.

In the reconstruction phase, the NN-FBP algorithm computes a reconstructed volume from limited projection data by combining multiple WBP reconstructions with different filters into a single reconstruction. A key ingredient of the algorithm is the application of a pixel-wise nonlinear scaling operation to each of the WBP images. Following this operation, the images are combined by taking a weighted sum of the scaled WBP images. As a final step, another nonlinear scaling operation is applied to this combined image (see reconstruction phase in Fig. 6.3).

Note that without these nonlinear scaling operations, the final reconstruction can also be obtained by first creating a weighted sum of the different filters, and performing a Weighted Backprojection with the resulting filter, as the WBP algorithm is a linear method with respect to the used filter. Because of this, such a method will not be able to produce more accurate reconstructions than standard Weighted Backprojection with an appropriately chosen filter. Also, because of the nonlinear scaling operation, it is not possible to directly compare the filters of the NN-FBP method with standard filters for WBP.

By using the nonlinear scaling operation, the NN-FBP algorithm is able to reduce the artifacts that are usually present in standard Weighted Backprojection reconstructions when only a small number of projections are available. An example image with standard Weighted Backprojection, a linear combination of two Weighted Backprojections, and a combination of two Weighted Backprojections with nonlinear scaling is shown in Fig. 6.2. As expected, the figure shows that the linear combination is identical to a single Weighted Backprojection reconstruction, while the combination with nonlinear scaling is significantly more accurate.

Learning phase

The question remains how the different filters and weights have to be chosen, such that the method produces accurate reconstructions. In Chapter 5, it is shown that ideas from artificial neural network theory can be used to find good filters and weights. Specifically, filters and weights can be learned by the NN-FBP method in a separate learning phase, in which the method is presented with high-quality reconstructions of a set of training objects. In artificial neural network theory, this technique is called supervised learning. In the learning phase, the filters and weights are iteratively adjusted until the NN-FBP reconstructions match the presented high-quality reconstructions. Afterwards, the trained filters and weights can be used to accurately reconstruct objects that are similar to the ones used for training, using only a limited number of projections. The angle distribution of the limited number of projections has to be specified during the learning phase, and the learned filters and weights will be specific to the chosen distribution. To reduce the influence of the specific angles that are chosen, NN-FBP uses angle-independent filters, i.e. the same filters are used for each projection. An important requirement of the NN-FBP method is that the reconstructed objects should consist only of materials that were also present in the training objects. When this requirement is satisfied, the NN-FBP method is able to produce accurate reconstructions, even for objects with different shapes and/or sizes as the training objects. A schematic overview of both the learning phase and subsequent reconstruction of the NN-FBP method is given in Fig. 6.3.

As opposed to previous advanced reconstruction methods, specific prior knowledge is not explicitly used in the NN-FBP method. Instead, the method learns to exploit certain characteristics of the training objects by adjusting the filters and weights appropriately. Because the exploited characteristics are learned automatically by the method, it has a broader applicability than previous advanced 3D reconstruction methods. Also, since NN-FBP is based on the efficient Weighted Backprojection algorithm, it is computationally efficient as well, enabling high throughput of 3D reconstructions. An additional advantage is that existing implementations of the Weighted Backprojection algorithm can be used to easily implement the NN-FBP method. A final advantage is that it is possible to include the segmentation step in the NN-FBP method by using segmented high-quality reconstructions of the training objects in the learning phase. In this case, the NN-FBP method will reconstruct objects with voxel values that are very close to their segmented value, and the final segmentation can be performed by simple rounding to the nearest segmented value. This removes the need for manual segmentation, which can be problematic for other methods when only a limited set of

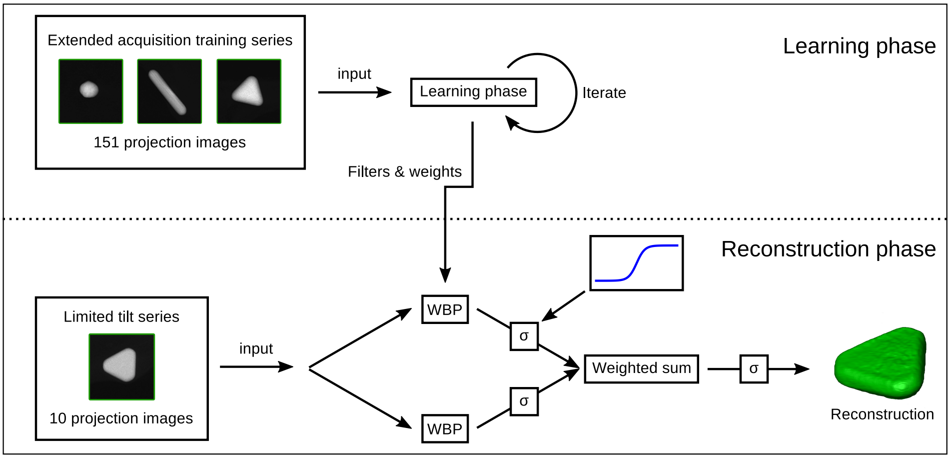


Figure 6.3: Schematic overview of the NN-FBP procedure. In the learning phase, the extended acquisition series are used as an input to learn filters and weights specific to the training objects. In the reconstruction phase, the learned filters are used in multiple WBP reconstructions with an additional pixel-wise nonlinear scaling operation, which are combined to obtain a single reconstruction of a limited tilt series.

projections is available.

6.3 Results

Qualitative results

In a first experiment, tilt series of a nanosphere, a nanorod and a nanotriangle are acquired over an angular tilt range of ± 75° with a tilt increment of 1°. These three series are used as training series, resulting in a set of filters that will be used during the NN-FBP approach. The resulting NN-FBP algorithm is applied to a limited tilt series that was acquired from a different nanotriangle. Although only 10 projection images obtained over a range of ± 75° are used during the NN-FBP reconstruction, it needs to be pointed out that we also acquired an extended series of 151 projection images. The SIRT reconstruction of the extended dataset was used as ground truth, in order to evaluate the NN-FBP outcome. Figure 6.4.a presents a volume rendering of this full range SIRT reconstruction. In all experiments, we used 200 iterations for the SIRT reconstructions, which was empirically verified to produce accurate reconstructions. The result of the NN-FBP algorithm is shown in Fig. 6.4.b. It must be stressed that in this case only 10 projection images were used. It can be seen that the 3D volume visualization of the NN-FBP reconstruction is in very good agreement with the SIRT reconstruction of the full data series. The top and side facet can clearly be distinguished in the corresponding orthoslices in Fig. 6.4.e,i,m and f,j,n. On the other hand, when comparing the SIRT reconstruction based on the extended series with the SIRT reconstruction based on 10 projection images (Fig. 6.4.c,g,k,o), it can

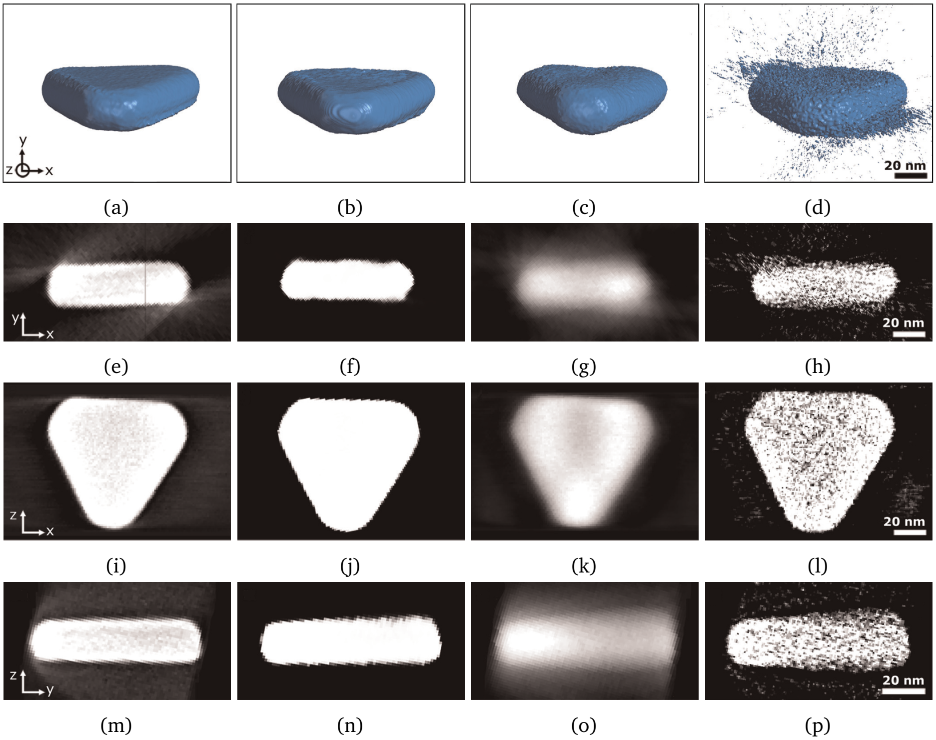


Figure 6.4: Reconstructed volumes of a nanotriangle using (a) the full dataset of 151 projections and the SIRT algorithm, and a limited dataset of only 10 projections using (b) the NN-FBP, (c) the FBP and (d) the SIRT algorithm. Xy, xz and yz orthoslices through the (e,i,m) full SIRT, (f,j,n) the NN-FBP, (g,k,o) the limited SIRT and (h,l,p) the limited WBP reconstructions of the nanotriangle.

be seen that the faceted shape is less pronounced. In the WBP reconstruction applied on 10 projection images (Fig. 6.4.d,h,l,p), severe noise and streaking artifacts can be distinguished. These artifacts can be prohibitive for further analysis of the scanned object, such as volume or shape calculations. Therefore, the WBP reconstruction will be left out in the further analysis. The benefits of NN-FBP become obvious; the number of images required for a 3D reconstruction using NN-FBP is reduced by a factor of 15, but the quality is comparable to a reconstruction based on a full data series with a tilt increment of 1°.

In Fig. 6.5 and Fig. 6.6, results for a nanosphere and a nanorod are presented, respectively. Here, the training of the filters was again obtained by 3 training series. For the nanosphere, extended series of the nanorod and both nanotriangles were used. The training step for the nanorod was performed by the extended series of the nanosphere and both nanotriangles. These nanostructures yield less facets and as a consequence, the general morphology as visualized in Fig. 6.5.b,c and Fig. 6.6.b,c appears to be better preserved when using only 10 projections. However, missing wedge artifacts can be clearly seen in the orthoslices presented in Fig. 6.5.f,i,l and Fig. 6.6.f,i,l. Because

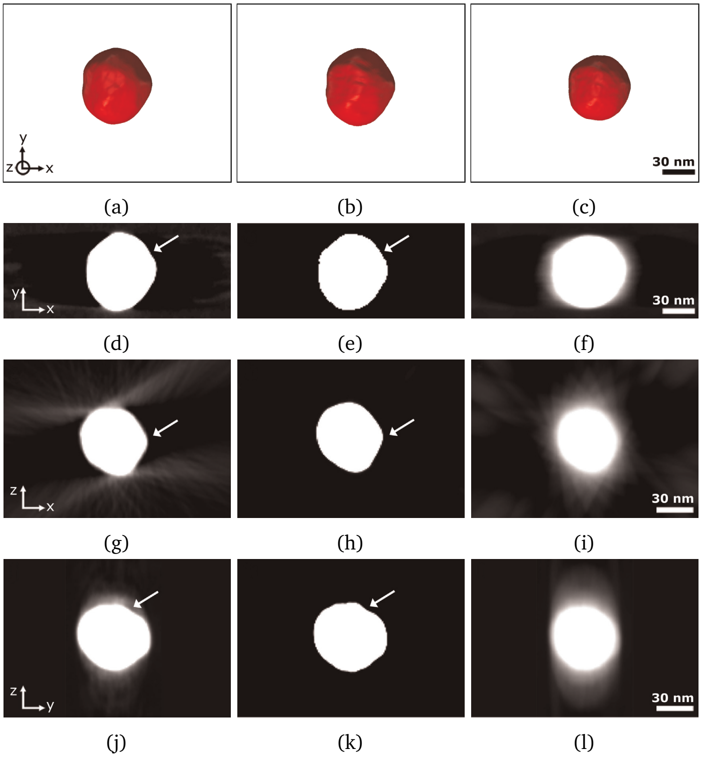


Figure 6.5: Reconstructed volumes of a nanosphere using (a) the full dataset of 151 projections and the SIRT algorithm, and a limited dataset of only 10 projections using (b) the NN-FBP and (c) the SIRT algorithm. Xy, xz and yz orthoslices through the (d,g,j) full SIRT, (e,h,k) the NN-FBP and (f,i,l) the limited SIRT reconstructions of the nanosphere. The white arrows indicate the presence of surface roughnesses. It is clear that these features are visible both in the orthoslices through the full SIRT and in the orthoslices through the NN-FBP reconstruction; however, in the limited SIRT reconstruction they are not detectable.

of such artifacts, some features of the morphology indicated by white arrows in both the orthoslices through the full SIRT reconstruction (Fig. 6.5.d,g,j) and the NN-FBP reconstruction (Fig. 6.5.e,h,k) are not clearly visible in the orthoslices through the limited SIRT reconstruction (Fig. 6.5.f,i,l).

Quantitative results

As a quantitative measure, a difference reconstruction for the nanosphere is constructed by subtracting the SIRT (Fig. 6.7.a) and NN-FBP reconstructions based on 10 projection images (Fig. 6.7.b) from the full SIRT reconstruction of the nanorod. The threshold value for the full SIRT reconstruction is obtained from the histogram. The histogram of the limited SIRT reconstruction, however, is largely influenced by the lack of projection images. In Fig. 6.8, comparisons are shown between the histograms of the full SIRT reconstruction and the limited SIRT reconstruction for each nanoparticle. Clearly, one

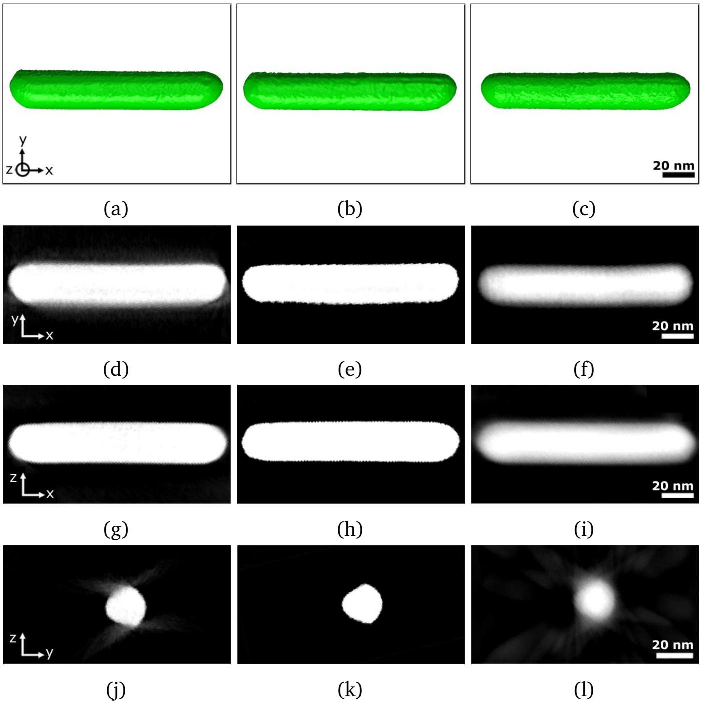


Figure 6.6: Reconstructed volumes of a nanorod using (a) the full dataset of 151 projections and the SIRT algorithm, and a limited dataset of only 10 projections using (b) the NN-FBP and (c) the SIRT algorithm. Xy, xz and yz orthoslices through the (d,g,j) full SIRT, (e,h,k) the NN-FBP and (f,i,l) the limited SIRT reconstructions of the nanorod.

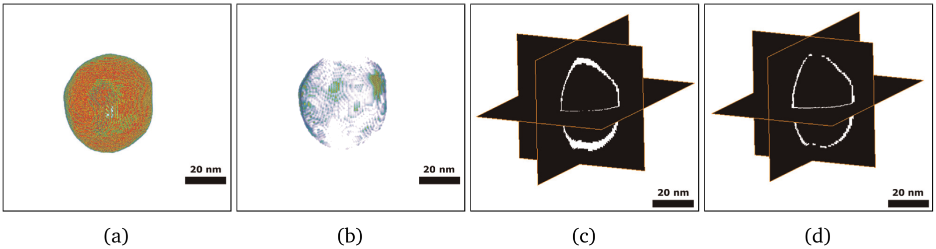
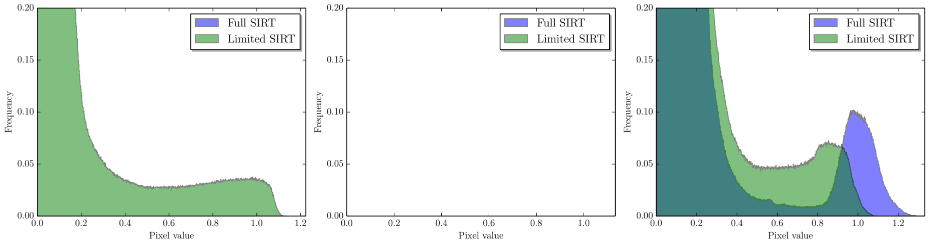
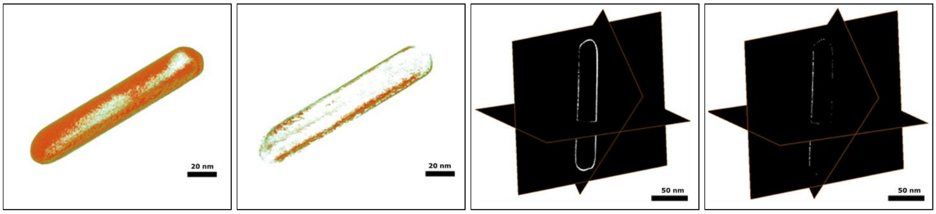


Figure 6.7: Difference reconstructions of the nanosphere constructed by subtracting (a) the SIRT and (b) NN-FBP reconstruction of 10 projection images from the full SIRT reconstruction representing the missing volume and its orthoslices (c) and (d), respectively. The volume misinterpretation for the NN-FBP reconstruction equals only 1.6%, which is indicated by the fine shell of the difference reconstruction. The thicker shell present in the difference reconstruction of the limited SIRT equals a volume misinterpretation of 21.5%.



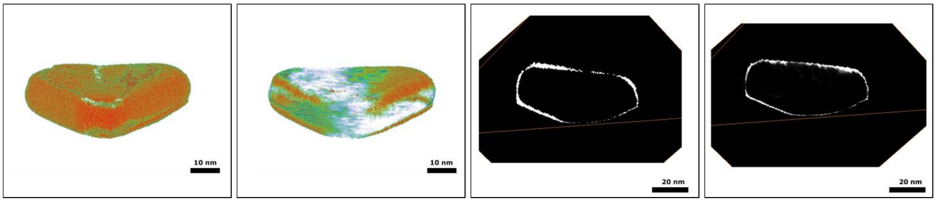
(a) (b) (c)

Figure 6.8: Histograms of SIRT reconstructions of the nanorod (a), the nanosphere (b), and the nanotriangle (c), with all 151 projections (Full), and only 10 projections (Limited). It is clear that the poor quality of the limited SIRT reconstruction hampers an objective choice of a threshold for segmentation purposes.



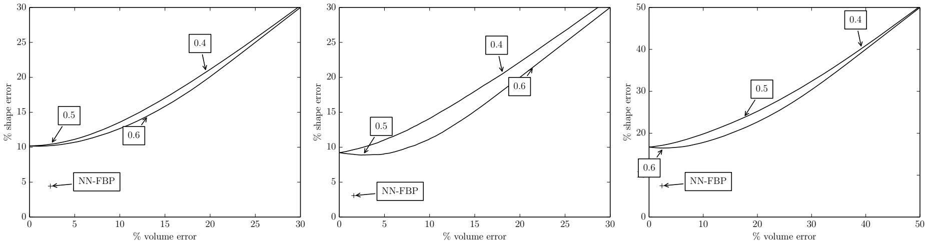
(a) (b) (c) (d)

Figure 6.9: Difference reconstructions of the nanorod constructed by subtracting (a) the SIRT and (b) NN-FBP reconstruction of 10 projection images from the full SIRT reconstruction representing the missing volume. Corresponding orthoslices through the difference reconstruction of a nanorod using (c) SIRT and (d) NN-FBP on 10 projections are shown. The volume misinterpretation for the NN-FBP reconstruction equals only 2.3%, which is indicated by the fine shell of the difference reconstruction. The thicker shell present in the difference reconstruction of the limited SIRT equals a volume misinterpretation of 13.1%.



(a) (b) (c) (d)

Figure 6.10: Representation of the shape misinterpretation, which for (a) the limited SIRT reconstruction shows a volume underestimation at the center of the nanotriangle and a volume overestimation at the tips of the nanotriangle. In (b) the shape misinterpretation for the limited NN-FBP reconstruction is visualized. (c) and (d) represent the orthoslices through the limited SIRT and NN-FBP reconstruction, respectively. The shape misinterpretation for the NN-FBP reconstruction equals 7.5%, which is indicated by the fine shell of the difference reconstruction. The thicker shell present in the difference reconstruction of the limited SIRT equals a shape misinterpretation of 16.5%.



(a) (b) (c)

Figure 6.11: Plots of the relative error in the shape and the volume of segmented limited SIRT reconstructions of the nanorod (a), the nanosphere (b), and the nanotriangle (c), compared to the full SIRT reconstruction, for different thresholds of the limited SIRT reconstruction. The error in shape is defined as the number of voxels that are labeled differently in the segmentations of the limited in comparison to the full SIRT reconstruction. The error in volume is defined as the absolute value of the difference between the volumes of the segmented limited SIRT reconstruction and the segmented full SIRT reconstruction. The errors for a threshold of 0.4, 0.5 and 0.6 are indicated with arrows and intermediate thresholds are shown by a line. The errors of the NN-FBP method for each nanoparticle are also shown. Note that for each nanoparticle, the error of the NN-FBP reconstruction is closer to the origin than the error of any thresholded SIRT reconstruction.

would have trouble choosing correct threshold values on the basis of the limited SIRT histograms. Therefore, the same threshold value as the full SIRT reconstruction is used for the limited SIRT reconstructions. Since the NN-FBP reconstructions are already segmented, no threshold value is needed for them. Both from the visualization in Fig. 6.7.a, as well as the corresponding orthoslices through the difference reconstruction in Fig. 6.7.c, the volume misinterpretation of the limited SIRT reconstruction is clearly detectable. The orthoslices through the limited SIRT difference reconstruction of the nanosphere show a thick white shell. Here, the larger amount of white pixels indicates a volume misinterpretation of 21.5% when using the SIRT algorithm on the dataset of only 10 projection images. From Fig. 6.7.b and its corresponding orthoslices in Fig. 6.7.d, it is clear that the volume reconstructed with NN-FBP on 10 projection images is close to the actual volume. The NN-FBP reconstruction has only 1.6% of volume underestimation. For the nanorod (Fig. 6.9), the volume for SIRT applied to a limited dataset results in an underestimation of 13.1%. The NN-FBP reconstruction leads to a misinterpretation of only 2.3%. For the nanotriangle, the volume misinterpretation for the limited SIRT reconstruction equals 2.7%. When reconstructing the 10 projection dataset with the NN-FBP algorithm, the volume misinterpretation equals 2.4%. For the nanotriangle, the volume misinterpretation of the limited SIRT reconstructions is close to the misinterpretation of the NN-FBP reconstruction. In this case, however, the volume misinterpretation of the limited SIRT reconstruction gives a misleading result, due to a volume underestimation at the center of the nanotriangle and a volume overestimation at the tips of the nanotriangle. In general, the volume misinterpretation can be misleading due to the canceling out of overestimation and underestimation. Clearly, the evaluation of the quality of the reconstruction can not only be based on an inspection of the volume error. Therefore, the shape error is introduced, which corresponds to the number of voxels that are labeled differently in the segmentations

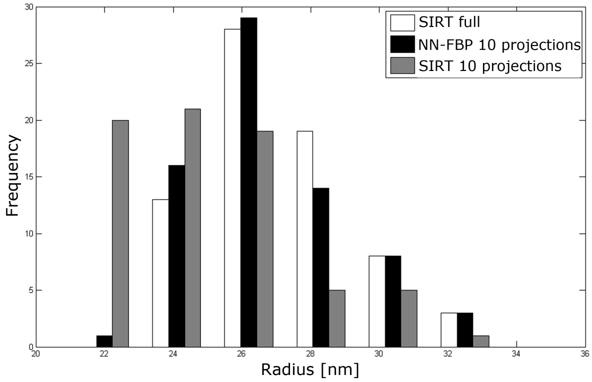


Figure 6.12: Distribution of the radii of nanospheres reconstructed using SIRT on full datasets of 151 projections (white), NN-FBP (black) and SIRT on limited datasets of 10 projections (gray). The distributions of SIRT full and NN-FBP 10 projections are in good agreement. When SIRT is applied on the limited datasets, a different distribution is found due to the misinterpretation of the volume.

of the limited data reconstructions in comparison to the full SIRT reconstruction. In this manner, both the local volume underestimation at the center as well as the volume overestimation at the tips is taken into account. For the nanotriangle, there is a 16.5% shape misinterpretation for the limited SIRT reconstruction (Fig. 6.10). The shape error for the NN-FBP reconstruction equals 7.5%, which is clearly smaller in comparison to the shape error of the limited SIRT reconstruction. An extended investigation of the influence of the chosen threshold value on the shape error and volume error of the limited SIRT reconstructions is shown in Fig. 6.11. Note that from Fig. 6.11, one can conclude that the errors depend heavily on the chosen threshold value, showing the difficulties one would have when choosing a threshold value both optimizing shape and volume error for limited SIRT reconstructions.

Statistical results

In general it is difficult to obtain statistical results when applying electron tomography. As pointed out previously, the acquisition of tilt series for electron tomography is very time consuming and a large electron dose is required in the case of small tilt increments. The NN-FBP algorithm is therefore of great interest as it can be applied to reduce the acquisition time. In this manner a large set of nanostructures can be investigated in an efficient manner, leading to statistical results. Using the NN-FBP approach explained above, training was performed on a set of 20 nanoparticles, and a total of 71 nanospheres was investigated. The number of nanoparticles to train on was chosen empirically, such that there were both enough particles to use in the learning phase, and enough particles to obtain statistical results from. In Fig. 6.12, the distribution of the radii of these nanospheres is evaluated. In order to investigate the reliability of the NN-FBP approach, extended tilt series of 151 images were acquired for all particles. The outcome of the NN-FBP algorithm and the SIRT algorithm, using only 10 projections, is then compared to the measurements based on the SIRT reconstruction using 151 projections. The distribution indicated in gray in Fig. 6.12 presents the radii distribution for the nanospheres reconstructed using SIRT applied to limited datasets and clearly gives a different distribution in comparison to the radii distribution of the full SIRT reconstruction, which is presented in white. The average radius found in this manner equals (24.1 ± 0.59) nm, which is significantly smaller than the actual radius which equals (27.1 ± 0.25) nm, found through the full SIRT reconstructions. As the optical properties, such as the absorption cross section, are dependent on the shape and size of the nanoparticles, it is of key importance to retrieve the real nanoparticle morphology. A small difference of a few nanometer can already influence the outcome of the optical response [LE00; Per+10]. The radii distribution of the NN-FBP reconstruction (black), however, is in good agreement with the results extracted from the full SIRT data (white). The average radius of the NN-FBP reconstructed nanospheres equals (26.8 ± 0.29) nm. This value is in good agreement with the actual average radius and shows a clear overlap of the error bars. It is again clear that the SIRT algorithm can not provide reliable information when limited datasets are investigated. These results confirm the reliability of the NN-FBP algorithm and demonstrate the possibility of combining electron tomography and statistical measurements.

6.4 Conclusion

We have shown that the NN-FBP reconstruction algorithm is able to yield electron tomography reconstructions based on highly limited data with a comparable quality to a reconstruction based on a full data series with a tilt increment of 1°. The decrease in acquisition time and the use of an efficient reconstruction method enables us to examine a broad range of nanostructures in a statistical manner. The NN-FBP algorithm also has promising prospects for the 3D investigation of beam sensitive samples, where only a limited amount of projection images need to be acquired.

7

Integrating TomoPy and the ASTRA toolbox

7.1 Introduction

In transmission X-ray tomography experiments performed at synchrotron facilities, large amounts of projection data are produced in a short time. Current detector technology allows collection of projections at kHz frame rate, enabling 3D imaging of dynamic systems [Gib+15], *in–situ* studies of materials [Pat+16], and monitoring the evolution of biological systems [Moo+13].

Processing these datasets in a time comparable with data collection is essential to properly capture the sample evolution and adjust the instrument settings during the experiment; this requires algorithms optimized for high-performance computing (HPC), which have to be easily available and usable by the beamline users. Furthermore, many advanced experiments, such as those with extremely high spatial or temporal resolutions [SA10; Mok+13] and of dose-sensitive objects [Lov+13], require a variety of pre-processing, post-processing, and reconstruction algorithms to reduce artifacts in the final reconstruction.

In this chapter, we present the integration of two Python toolboxes which, together, allow users to easily apply advanced tomographic algorithms on large-scale experimental datasets in an efficient way: the TomoPy toolbox [Gür+14] and the

---

This chapter is based on:

D. M. Pelt, D. Gürsoy, W. J. Palenstijn, J. Sijbers, F. De Carlo, and K. J. Batenburg. "Integration of TomoPy and the ASTRA toolbox for advanced processing and reconstruction of tomographic synchrotron data". *Journal of Synchrotron Radiation* (Submitted for publication).

ASTRA toolbox [Aar+15]. By combining both toolboxes, we are able to leverage the advantages of both to create an improved workflow for beamline users.

The TomoPy toolbox is specifically designed to be easy to use and deploy at a synchrotron facility beamline. It supports reading many common synchrotron data formats from disk [De +14], and includes several pre-processing and post-processing algorithms commonly used for synchrotron data. TomoPy also includes several reconstruction algorithms, which can be run on multi-core workstations and large-scale computing facilities. The algorithms in TomoPy are all CPU-based, however, which can make them prohibitively slow in the case of iterative methods, which are often required for advanced tomographic experiments.

The ASTRA toolbox provides highly efficient tomographic reconstruction methods by implementing them on Graphic Processing Units (GPUs). It includes advanced iterative methods and allows for very flexible scanning geometries. The ASTRA toolbox also includes building blocks which can be used to develop new reconstruction methods, allowing for easy and efficient implementation and modification of advanced reconstruction methods. However, the toolbox is only focused on reconstruction, and does not include pre-processing or post-processing methods that are typically required for correctly processing synchrotron data. Furthermore, no routines to read data from disk are provided by the toolbox.

By integrating the ASTRA toolbox in the TomoPy framework, the optimized GPU-based reconstruction methods become easily available for synchrotron beamline users, and users of the ASTRA toolbox can more easily read data and use TomoPy's pre-processing and post-processing methods.

This chapter is structured as follows: in Section 7.2, we give a more detailed explanation of TomoPy and the ASTRA toolbox, and explain how we integrated them. In Section 7.3, we give general instructions on how to install and use the combined toolboxes in practice. An example for a specific dataset is given in Section 7.4, and we conclude the chapter in Section 7.5.

7.2 Integrating TomoPy and the ASTRA toolbox

TomoPy

TomoPy is an open-source Python toolbox to perform tomographic data processing and image reconstruction tasks, developed at the Advanced Photon Source of Argonne National Laboratory [Gür+14]. The aim of the toolbox is to provide a high-level interface for data analysis and tomographic reconstruction of datasets at synchrotron light sources. TomoPy relies on standard scientific packages like NumPy, SciPy, and Scikit, and offers a free, open-source, modular, readable and manageable framework that researchers can use and contribute to easily. Python also offers easy integration with C or Fortran code through shared libraries in situations where computation speed is critical. In addition, the native control software running at several synchrotron facilities, EPICS[1], is accessible via Python[2], allowing simultaneous data analysis and real-time feedback on the instrumentation status. So far, TomoPy has been employed in reconstructions for various techniques from micro-CT [Duk+15] to X-ray fluorescence tomography [Gür+15b], X-ray scattering tomography [Gür+15a], Lorentz electron microscopy [PG15] and deployed on large scale computing facilities [Biç+15].

TomoPy includes a plethora of processing functions from pre-processing to image reconstruction of synchrotron tomographic data. It includes ring removal algorithms, such as generalized Titarenko's algorithm [Miq+14], the Fourier Wavelet approach [Mün+09], and recently, the ring correction based on median filters [MWC15]. The estimation of rotation center can be calculated using the image entropy calculation based method [DBS06] or Vo's Fourier method [Vo+14]. A single-step X-ray phase retrieval algorithm based on Paganin filtering is also available for phase-contrast datasets [Pag+02].

In addition to Gridrec [Dow+99], which is the traditionally used analytical image reconstruction algorithm, TomoPy also offers variants of algebraic reconstruction methods (ART, BART, SIRT), and maximum-likelihood expectation maximization (ML-EM) approaches, as well as their regularized variations (PML). Ordered-subset implementations of all algorithms are also available for efficient calculations, for example, the well-known ordered-subset expectation maximization (OSEM) algorithm. Another important property of TomoPy is that it provides X-ray matter interaction simulation tools, such as X-ray transmission or wave propagation, that can be used to evaluate efficiency of various coding scenarios or as a platform for modeling. TomoPy algorithms are also suitable for grid-computing and massive parallelization when needed. Experiments with iterative algorithms and large tomography datasets show that TomoPy iterative methods can scale up to 8K cores on an IBM BG/Q supercomputer with almost perfect speedup and can reduce total reconstruction times for large datasets by more than 95.4% on 32K cores relative to 1K cores. Moreover, the average reconstruction times are improved from 2 hours (256 cores) to 1 minute (32K cores), thus enabling near-real-time use [Biç+15].

The ASTRA toolbox

The *ASTRA toolbox* is an open-source software toolbox developed at the University of Antwerp, Belgium, and at the Centrum Wiskunde Informatica (CWI), Amsterdam, The Netherlands, that is focused on the reconstruction of 2D and 3D tomographic datasets [Aar+15]. The aim of the toolbox is to provide a fast and flexible development platform for tomographic reconstruction algorithms. Because of its flexibility, it can be applied to various scanning geometries and acquisition modes, such as (bio)medical and industrial μCT [Pla+15], electron tomography [Roe+12], neutron tomography [PL13; Van+15], and synchrotron tomography [Rei+13]. The toolbox uses Graphics Processing Units (GPUs) to perform accelerated parallel computations, reducing the computation time of many tomographic operations [PBS11]. Most 2D operations can also be run on

---

[1] `http://www.aps.anl.gov/epics`

[2] `http://pyepics.github.io/pyepics/`

	Method	CPU	GPU		Method	CPU	GPU
TomoPy	ART	×		ASTRA	ART	×	
	BART	×			BP	×	×
	Gridrec	×			CGLS	×	×
	MLEM	×			FP	×	×
	OSEM	×			FBP	×	×
	PML	×			MLEM		×
	OSPML	×			SART	×	×
	SIRT	×			SIRT	×	×

Table 7.1: List of tomographic reconstruction methods included in TomoPy and ASTRA for two-dimensional parallel-beam geometries.

standard CPUs, in which case the toolbox supports different projection kernels, i.e. ways of discretizing the projection operations. A comparison of various projection kernels can be found in [XM06]. Through a MATLAB and Python interface, the tomographic operations can be easily used and combined with advanced numerical code. The toolbox also provides a matrix-like interface to linear tomography operators, allowing them to be easily used in existing and new code [Ble+15].

The ASTRA toolbox includes many popular tomographic reconstruction methods (see Table 7.1), such as the analytic filtered backprojection (FBP) method and the iterative SIRT and CGLS methods. These methods support various parameters that can help improve reconstruction quality, for example the choice of filter to use in the FBP method, and additional nonnegativity constraints in the SIRT method. An important feature of the ASTRA toolbox is that it also provides *building blocks* that can be used to develop advanced tomographic reconstruction methods. For example, using the optimized methods for the forward projection of objects and the backprojection of sinograms, it is possible to develop efficient advanced iterative methods, such as total variation regularized methods, using the ASTRA toolbox. A recent addition to the toolbox is a *plugin* system, which enables algorithm developers to easily distribute new tomographic reconstruction methods, and ASTRA users to easily install and use them with minimal changes in production scripts. With the TomoPy integration presented in this chapter, these ASTRA plugins will automatically be usable in TomoPy as well.

Implementation

The code to integrate TomoPy and the ASTRA toolbox is written in the Python language, since TomoPy is mainly written in Python and the ASTRA toolbox includes a Python interface as well. Specifically, a first step was to add code to TomoPy that enables the use of other Python libraries to perform tomographic reconstruction instead of TomoPy's included algorithms. Using this new feature, code was added which enables the use of the ASTRA toolbox to perform the reconstruction. Note that other tomographic reconstruction libraries that include a Python interface can be integrated in TomoPy in the same way.

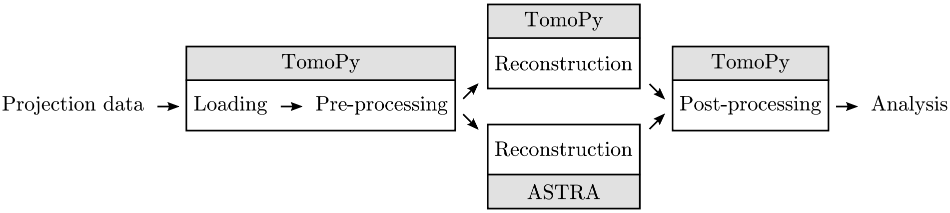


Figure 7.1: Schematic overview of the workflow of processing a dataset with the integrated TomoPy and ASTRA framework. Note that the reconstruction step can be performed by either TomoPy or the ASTRA toolbox.

In the interfacing code, the geometry defined by TomoPy (i.e., the number of detector pixels, the angles for which projections are acquired, and the center of rotation) is translated to a corresponding ASTRA geometry, and the chosen ASTRA reconstruction method is performed. Afterwards, the result of the reconstruction is stored in TomoPy memory, and all ASTRA objects are cleaned up. In this way, the reconstruction step is completely self-contained and independent of any pre-processing or post-processing step. An advantage of this independence is that user scripts do not have to be rewritten to use the ASTRA toolbox: only the reconstruction function call has to be modified. Also, changing between different reconstruction methods, between CPU and GPU implementations, and between different reconstruction parameters usually only requires changes in a single line of the user script. Examples of these minimal changes are shown in Section 7.3. A schematic overview of the full processing workflow from loading the raw data to analysis is shown in Fig. 7.1.

7.3 Installation and usage

Installation

Both TomoPy and the ASTRA toolbox can be installed using the Conda package management system.[3] The advantage of using Conda over other Python package management systems is that Conda allows for the inclusion of non-Python library dependencies, which are commonly needed for numerical toolboxes such as TomoPy and the ASTRA toolbox. Since the goal of both toolboxes is to be easily installable at the various workstations and computational clusters available at synchrotrons, which may each be running a different environment of installed libraries and library versions, the ability to tightly control the library dependencies is important to create a user-friendly installation process. To install TomoPy in a Conda environment, the following command can be used:

```
$ conda install tomopy \
        -c https://conda.anaconda.org/dgursoy
```

A similar command can be used to install the ASTRA toolbox:

---

[3] `http://conda.pydata.org`

```
$ conda install astra-toolbox \
        -c https://conda.anaconda.org/astra-toolbox
```

Note that both toolboxes can be installed and used independently from each other. TomoPy will automatically detect whether the ASTRA toolbox is available, and enables the use of ASTRA methods if this is the case. Both toolboxes can also be compiled and installed from source code, which can be downloaded from their respective git repositories.[4,5] Compared to installing using Conda, it is easier to make modifications and contribute to the development of the toolboxes when installing from source code, but the compilation step requires the availability of several library dependencies on the workstation.

Usage

We will now show how the new features can be used after installation of both toolboxes. The following example script shows a simple standard TomoPy workflow, loading data from disk, normalizing the data using the flatfield and darkfield images, and finally reconstructing with the standard TomoPy gridrec method:

```
 1 import tomopy
 2 
 3 # Read data from APS 32ID beamline
 4 prj, flat, dark =
 5     tomopy.exchange.read_aps_32id('data.h5')
 6 
 7 # Normalize data
 8 prj = tomopy.normalize(prj, flat, dark)
 9 
10 # Define acquired angles
11 ang = tomopy.angles(1024)
12 
13 # Reconstruct data using TomoPy gridrec
14 rec = tomopy.recon(prj, ang, algorithm='gridrec')
```

To modify this script to reconstruct using the ASTRA toolbox instead, only line 14 has to be changed, replacing `gridrec` with `tomopy.astra`, specifying which method to reconstruct with in the `method` option, and specifying which type of projection kernel to use in the `proj_type` option. An overview of common options that are used when reconstructing with the ASTRA toolbox is given in Table 7.2. For example, to reconstruct with FBP using a voxel-driven kernel, we change the final part of the above script to:

```
13 # Reconstruct data using ASTRA FBP
14 rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
15     options={'method': 'FBP', 'proj_type': 'linear'})
```

---

[4] https://github.com/tomopy/tomopy

[5] https://github.com/astra-toolbox/astra-toolbox

Option	Description	Example values
`'method'`	Which reconstruction method to use	`'FBP_CUDA'`, `'SIRT'`, `'CGLS_CUDA'`, ...
`'proj_type'`	Which projection kernel to use	`'linear'`, `'strip'`, `'cuda'`, ...
`'num_iter'`	Number of iterations to use in iterative method	`10`, `100`, `200`, ...
`'extra_options'`	Python dictionary with extra method-specific options	method-specific
`'gpu_list'`	List of GPU indices to use for reconstruction	`[0, 1, 2, 3]`, ...

Table 7.2: List of common options that are used when reconstructing with the ASTRA toolbox through TomoPy.

When running on a machine with a GPU with CUDA capabilities, the same reconstruction can be performed using optimized GPU code, greatly decreasing the needed computation time. This can be realized by specifying `cuda` as the projection kernel, and use a GPU-enabled method (`FBP_CUDA`):

```
# Reconstruct data using ASTRA FBP on the GPU
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options={'method': 'FBP_CUDA', 'proj_type': 'cuda'})
```

Iterative methods can be used by specifying the corresponding ASTRA method (e.g. `CGLS_CUDA` for a GPU-enabled CGLS method), and the number of iterations to use in the `num_iter` option:

```
# Reconstruct data using ASTRA CGLS on the GPU
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options={'method': 'CGLS_CUDA', 'proj_type': 'cuda',
    'num_iter': 10})
```

Most reconstruction methods in the ASTRA toolbox support several parameters that can help improve reconstruction quality. In the TomoPy integration, these parameters are specified by supplying them in the `extra_options` setting. For example, to add a nonnegativity constraint to the GPU-enabled SIRT method, we add `'MinConstraint':0` to the `extra_options` setting (note that lower bounds other than zero can also be used):

```
# Reconstruct data using ASTRA SIRT on the GPU,
# with nonnegativity constraint
extra_options = {'MinConstraint':0}
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options={'method': 'SIRT_CUDA', 'proj_type': 'cuda',
    'num_iter': 100, 'extra_options': extra_options})
```

An overview of the various parameters that are supported by the reconstruction methods can be found on the website of the ASTRA toolbox.[6]

---

[6] `http://sourceforge.net/p/astra-toolbox/wiki/Home/`

If multiple GPUs are installed in the workstation running TomoPy and the ASTRA toolbox, the computations can be distributed over multiple GPUs by specifying a list of GPU indices in the `gpu_list` option. Since each slice of the reconstruction can be computed independently from the other slices, a significant reduction of computation time can be achieved by distributing the computations in this way. For example, to use four installed GPUs, labeled 0 through 3, we use `[0, 1, 2, 3]` as the `gpu_list`:

```
# Reconstruct data using ASTRA CGLS on 4 GPUs
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options={'method': 'CGLS_CUDA', 'proj_type': 'cuda'
    ,'num_iter': 10, 'gpu_list': [0, 1, 2, 3]})
```

Finally, ASTRA plugins can be used by first registering them with the ASTRA toolbox itself, and using the method name defined by the plugin as the `method` option. Extra parameters for the reconstruction can be specified using the `extra_options` setting, similar to standard ASTRA methods. An ASTRA plugin is typically distributed as a Python class within a Python package, which has to be imported separately. After importing, the `astra.plugin.register` method is used to register a plugin with the ASTRA toolbox. For example, suppose that there is a plugin class `tvtomo.plugin` within the `tvtomo` package, with the method name `TV-FISTA` and an additional parameter `tv_reg`. To use this plugin in TomoPy, the following code can be used:

```
# Reconstruct data using an ASTRA plugin,
# performing FISTA total variation minimization
import astra
import tvtomo
astra.plugin.register(tvtomo.plugin)
extra_options = {'tv_reg':1}
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options={'method': 'TV-FISTA', 'proj_type': 'cuda',
    'num_iter': 100, 'extra_options':extra_options})
```

Computation time

In Fig. 7.2, a comparison is shown between the computation times per slice of reconstructions computed with different methods of both TomoPy and the ASTRA toolbox, for a single slice of a dataset with 1200 detector pixels and 1024 projections. All reconstructions were computed on a workstation with two Intel Xeon E5-2623 v3 CPUs (four cores each) and two Geforce GTX Titan Z cards (two GPUs each), running the Fedora 21 operating system. We compare the computation times of gridrec computed using TomoPy, FBP computed using the ASTRA toolbox, and 100 iterations of SIRT computed using both TomoPy and the ASTRA toolbox. For the CPU-based methods of TomoPy and the ASTRA toolbox, computation times are shown for using both a single core and all eight cores of the machine. For the GPU-based methods of the ASTRA toolbox, computation times are shown for using both a single GPU and all four installed GPUs.

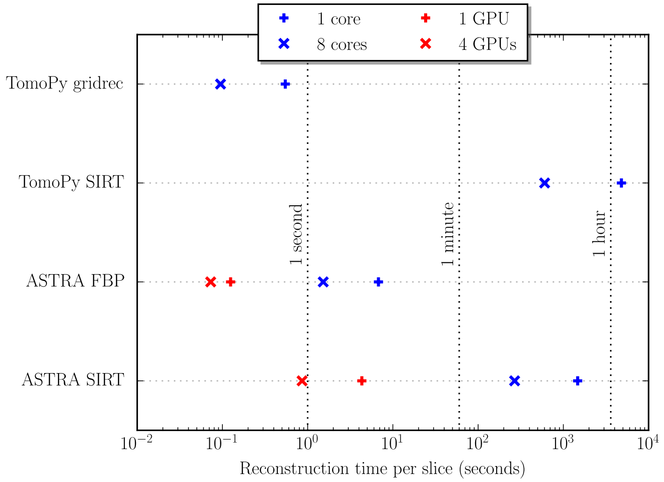


Figure 7.2: Computation time per slice of reconstructing with the gridrec method computed with TomoPy, the FBP method computed with the ASTRA toolbox, and 100 iterations of the SIRT method computed with both, for 1024 projections and a detector width of 1200 pixels. Results are shown for using a single CPU core, 8 CPU cores, a single GPU, and 4 GPUs.

The results of Fig. 7.2 show that for the SIRT method, a significant reduction of computation time can be achieved by using GPUs instead of CPUs for computation, with the computation time per slice when using TomoPy and eight CPU cores is roughly 700 times the computation time when using the ASTRA toolbox and four GPUs. Note that this reduction of computation time can be important in practice, since in this case, computing the SIRT reconstruction of 200 slices would take roughly 3 minutes with four GPUs, compared to more than 33 hours with eight CPU cores. Therefore, by using GPU-based methods, it is possible to compute iterative reconstructions during experiments, enabling direct inspection of the reconstructions and the possibility of making adjustments to improve the experimental results during the experiment itself.

In contrast to the iterative SIRT method, TomoPy's CPU-based gridrec method takes approximately the same time to compute as the GPU-based FBP method. This is expected for problems with a relatively large number of projections, since the most costly computations of the gridrec method are the 2D Fourier transforms, for which the computation time is independent of the number of projections. On the other hand, in the FBP method the most costly computation is the backprojection operation, for which the computation time scales linearly with the number of projections. Note that the reconstruction quality of gridrec and FBP reconstructions are usually similar [MS12].

7.4 Example

In this section, we give an example of the full processing workflow of reconstructing a tomographic synchrotron dataset, acquired at the 32-ID beamline of the Advanced

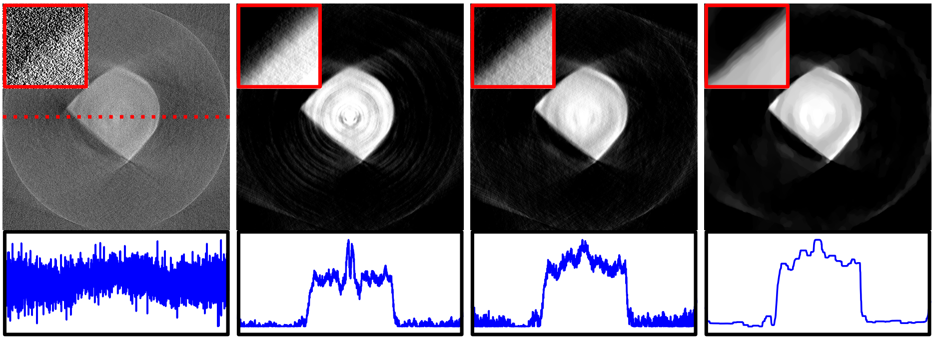

(a) (b) (c) (d)

Figure 7.3: Reconstructions of a single slice of a sample in a high pressure diamond anvil cell which blocks 86 of the 359 projections over 180°, with 2560 detector pixels per projection. Reconstructions are computed with (a) gridrec (TomoPy), (b) and (c) SIRT with a nonnegativity constraint (ASTRA), and (d) TV-minimization using FISTA (ASTRA plugin). In (a), (c), and (d), a ring-removal pre-processing step [Mün+09] was applied using TomoPy. A line profile of the center, indicated in (a) by a dotted line, is shown for each reconstruction, as well as a cropped 128 × 128 pixel section of the upper left part of the sample.

Photon Source of Argonne National Laboratory. We compare the resulting reconstructions of a single slice using various reconstruction methods, both with and without ring-removal pre-processing applied. The dataset is of a sample under pressure in a diamond anvil cell, whose frame blocks part of the acquired projections, rendering them unusable. The result is a limited-angle tomographic problem, where the acquired projections do not span the entire 180° range. Specifically, projections of 2160 × 2560 pixels were acquired in 0.5° intervals over a 137° range, for a total of 273 projections. It is typically difficult to obtain accurate reconstructions for limited-angle problems, with standard methods producing wedge artifacts in the direction of the missing projection angles [DB98].

In Fig. 7.3, reconstructions are shown of a single slice of the sample, reconstructed with various reconstruction methods. In each reconstruction except for the one shown in Fig. 7.3b, a ring-removal pre-processing method that is included in TomoPy [Mün+09] was used to suppress ring artifacts. In Fig. 7.3a, the reconstruction computed with TomoPy's gridrec method is shown. In Figs. 7.3b and 7.3c, ASTRA's GPU-enabled SIRT method was used to compute the reconstructions. Finally, a reconstruction regularized with total variation minimization is shown in Fig. 7.3d, computed using an ASTRA plugin that implements the FISTA method [BT09a]. The Python script used to compute the reconstruction of Fig. 7.3c is given below. Note that the scripts used to compute the other reconstructions are similar, only requiring minimal changes like the ones given in Section 7.3.

```python
import tomopy
import numpy as np

# Set up dataset variables
file_name = './data.h5'
```

```
center = 1286 # center of rotation
start = 740 # first slice
end = 1500 # last slice
miss_ang = [142, 228] # blocked angle range

# Read data from APS 32ID beamline
prj, flat, dark =
    tomopy.exchange.read_aps_32id(file_name,
    sino=(start, end))

# Remove the blocked projections
ang = tomopy.angles(359)
prj = np.concatenate((prj[:miss_ang[0]], prj[miss_ang[1]:]),
      axis=0)
ang = np.concatenate((ang[:miss_ang[0], ang[miss_ang[1]:]))

# Normalize data
prj = tomopy.normalize(prj, flat, dark)

# Remove ring artifacts
prj = tomopy.remove_stripe_fw(prj)

# Setup ASTRA options
opts = {}
opts['method'] = 'SIRT_CUDA'
opts['num_iter'] = 100
opts['proj_type'] = 'cuda'
opts['extra_options'] = {'MinConstraint':0}

# Perform reconstruction
rec = tomopy.recon(prj, ang, algorithm=tomopy.astra,
    options=opts)

# Export reconstructed slices as TIFFs
tomopy.io.writer.write_tiff_stack(rec, fname='./recs',
    start=start)
```

The results of Fig. 7.3 show that the gridrec reconstruction includes large amounts of noise artifacts, especially visible in the line profile, as well as wedge artifacts resulting from the missing projection angles. These artifacts can make further analysis, such as volume estimation, difficult or impossible, even with further post-processing. The iterative reconstructions, which can be computed efficiently using the GPU-enabled methods of the ASTRA toolbox, include less artifacts, significantly reducing noise in the reconstructed image. Note, however, that significant ring artifacts are present in Fig. 7.3b, where we did not use one of TomoPy's ring-removal pre-processing methods. In the reconstruction of Fig. 7.3c, the ring artifacts are significantly reduced, which shows that TomoPy's advanced pre-processing methods can be used to improve the reconstruction quality of ASTRA's reconstruction methods. Finally, the total variation minimization reconstruction of Fig. 7.3d shows that advanced regularized reconstruction methods can be distributed as ASTRA plugins and be used in combination with TomoPy to minimize artifacts in the final reconstruction.

7.5 Conclusions

In this chapter, we presented the integration of two Python toolboxes used for processing tomographic data: TomoPy and the ASTRA toolbox. The integration allows for combining the advanced I/O, pre-processing, and post-processing capabilities of TomoPy with the advanced tomographic reconstruction methods of the ASTRA toolbox. One advantage of the integration is that it enables the use of GPU-enabled methods included in the ASTRA toolbox to significantly improve computation time, especially for iterative reconstruction methods. Another advantage is that advanced iterative methods can be written and distributed as ASTRA plugins and subsequently used within TomoPy. Code has been added to TomoPy that automatically creates the necessary ASTRA objects, cleaning them up after computation has finished. As a result, only minimal changes are needed in user scripts to use the ASTRA toolbox within TomoPy.

We have shown how to install both toolboxes on a single machine, and how to use the various features of the integrated software. In particular, we have shown how to adjust an existing TomoPy script to reconstruct with the ASTRA toolbox, how to change between different ASTRA reconstruction methods and between CPU and GPU implementations, and how to specify options for each method. Furthermore, an example was given where an ASTRA plugin was used to reconstruct the acquired data. For a specific dataset, we compared the computation time of various methods included in TomoPy and the ASTRA toolbox, which showed that the GPU-enabled iterative SIRT method of the ASTRA toolbox significantly reduced computation time compared to the CPU-based SIRT method of TomoPy. Finally, we computed reconstructions using different reconstruction methods for a single slice of experimental data, showing how ASTRA's advanced reconstruction methods in combination with TomoPy's advanced pre-processing methods can help reduce artifacts in reconstructions of tomographic synchrotron data, in particular in challenging scenarios where only a limited set of projections are available.

Bibliography

[Aar+15] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers. "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography". *Ultramicroscopy* 157 (2015), pp. 35–47 (cit. on pp. 10, 54, 108, 109).

[AB09] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009 (cit. on p. 72).

[Bar+08] M. Bar Sadan, L. Houben, S. G. Wolf, A. Enyashin, G. Seifert, R. Tenne, and K. Urban. "Toward atomic-scale bright-field electron tomography for the study of fullerene-like nanostructures". *Nano letters* 8.3 (2008), pp. 891–896 (cit. on p. 94).

[BB00] S. Basu and Y. Bresler. "$\mathcal{O}(N^2 \log_2 N)$ filtered backprojection reconstruction algorithm for tomography". *IEEE Transactions on Image Processing* 9.10 (2000), pp. 1760–1773 (cit. on pp. 21, 77).

[Biç+15] T. Biçer, D. Gürsoy, R. Kettimuthu, F. De Carlo, G. Agrawal, and I. T. Foster. "Rapid Tomographic Image Reconstruction via Large-Scale Parallelization". English. In: *Euro-Par 2015: Parallel Processing*. Ed. by J. L. Träff, S. Hunold, and F. Versaci. Vol. 9233. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2015, pp. 289–302 (cit. on p. 109).

[Bjö96] A. Björck. *Numerical methods for least squares problems*. Siam, 1996 (cit. on pp. 17, 46).

[BK06] K. J. Batenburg and W. A. Kosters. "A neural network approach to real-time discrete tomography". In: *Combinatorial Image Analysis*. Springer, 2006, pp. 389–403 (cit. on pp. 19, 67, 78).

[Bla+15] E. Bladt, D. M. Pelt, S. Bals, and K. J. Batenburg. "Electron tomography based on highly limited data using a neural network reconstruction technique". *Ultramicroscopy* 158 (2015), pp. 81–88 (cit. on pp. 93, 131).

[Ble+15] F. Bleichrodt, T. Leeuwen, W. J. Palenstijn, W. Aarle, J. Sijbers, and K. J. Batenburg. "Easy implementation of advanced tomography algorithms using the ASTRA toolbox with Spot operators". *Numerical Algorithms* 71.3 (2015), pp. 673–697 (cit. on p. 110).

[BP12] K. J. Batenburg and L. Plantagie. "Fast Approximation of Algebraic Reconstruction Methods for Tomography". *IEEE Transactions on Image Processing* 21.8 (2012), pp. 3648–3658 (cit. on pp. 12, 13, 36, 38, 40, 43).

[BS11] K. J. Batenburg and J. Sijbers. "DART: a practical reconstruction algorithm for discrete tomography". *IEEE Transactions on Image Processing* 20.9 (2011), pp. 2542–2553 (cit. on pp. 8, 42, 66, 95).

[BSC01] H. M. J. Boffin, D. Steeghs, and J. Cuypers. *Astrotomography: indirect imaging methods in observational astronomy*. Vol. 573. Springer Science & Business Media, 2001 (cit. on p. 2).

[BT09a] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202 (cit. on pp. 8, 47, 66, 81, 116).

[BT09b] A. Beck and M. Teboulle. "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems". *IEEE Transactions on Image Processing* 18.11 (2009), pp. 2419–2434 (cit. on pp. 52, 54).

[Buz08] T. M. Buzug. *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*. Springer, 2008 (cit. on pp. 11, 41, 65).

[Car+10] E. Carbó-Argibay, B. Rodríguez-González, S. Gómez-Graña, A. Guerrero-Martínez, I. Pastoriza-Santos, J. Pérez-Juste, and L. M. Liz-Marzán. "The Crystalline Structure of Gold Nanorods Revisited: Evidence for Higher-Index Lateral Facets". *Angewandte Chemie* 122.49 (2010), pp. 9587–9590 (cit. on p. 93).

[CDK70] R. A. Crowther, D. J. DeRosier, and A. Klug. "The reconstruction of a three-dimensional structure from projections and its application to electron microscopy". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 317. 1530. The Royal Society. 1970, pp. 319–340 (cit. on p. 95).

[Cic+95] A. Cichocki, R. Unbehauen, M. Lendl, and K. Weinzierl. "Neural networks for linear inverse problems with incomplete data especially in applications to signal and image reconstruction". *Neurocomputing* 8.1 (1995), pp. 7–41 (cit. on p. 66).

[Cie08] R. Cierniak. "A 2D approach to tomographic image reconstruction using a Hopfield-type neural network". *Artificial intelligence in medicine* 43.2 (2008), pp. 113–125 (cit. on p. 66).

[Cie09] R. Cierniak. "New neural network algorithm for image reconstruction from fan-beam projections". *Neurocomputing* 72.13 (2009), pp. 3238–3244 (cit. on p. 66).

[CL14] R. Cao-Milán and L. M. Liz-Marzán. "Gold nanoparticle conjugates: recent advances toward clinical applications". *Expert opinion on drug delivery* 11.5 (2014), pp. 741–752 (cit. on p. 93).

[CP11] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145 (cit. on p. 47).

[DB98] A. H. Delaney and Y. Bresler. "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography". *IEEE Transactions on Image Processing* 7.2 (1998), pp. 204–221 (cit. on p. 116).

[DBS06] T. Donath, F. Beckmann, and A. Schreyer. “Automated determination of the center of rotation in tomography data”. *Journal of the Optical Society of America A* 23.5 (2006), pp. 1048–1057 (cit. on p. 109).

[DDD04] I. Daubechies, M. Defrise, and C. De Mol. “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint”. *Communications on pure and applied mathematics* 57.11 (2004), pp. 1413–1457 (cit. on p. 47).

[De +14] F. De Carlo, D. Gürsoy, F. Marone, M. Rivers, D. Y. Parkinson, F. Khan, N. Schwarz, D. J. Vine, S. Vogt, S.-C. Gleber, et al. “Scientific data exchange: a schema for HDF5-based storage of raw and analyzed data”. *Journal of Synchrotron Radiation* 21.6 (2014), pp. 1224–1230 (cit. on p. 108).

[DKM07] D. C. Durairaj, M. C. Krishna, and R. Murugesan. “A neural network approach for image reconstruction in electron magnetic resonance tomography”. *Computers in Biology and Medicine* 37.10 (2007), pp. 1492–1501 (cit. on p. 67).

[Dow+99] B. A. Dowd, G. H. Campbell, R. B. Marr, V. V. Nagarkar, S. V. Tipnis, L. Axe, and D. P. Siddons. “Developments in synchrotron x-ray computed microtomography at the National Synchrotron Light Source”. In: *Developments in X-Ray Tomography II*. Vol. 3772. 1999, pp. 224–236 (cit. on p. 109).

[Duk+15] D. J. Duke, A. B. Swantek, N. M. Sovis, F. Z. Tilocco, C. F. Powell, A. L. Kastengren, D. Gürsoy, and T. Biçer. “Time-resolved X-ray Tomography of Gasoline Direct Injection Sprays”. *SAE International Journal of Engines* 9.1 (2015), pp. 151–161 (cit. on p. 109).

[Erc+06] P. Ercius, M. Weyland, D. A. Muller, and L. M. Gignac. “Three-dimensional imaging of nanovoids in copper interconnects using incoherent bright field tomography”. *Applied physics letters* 88.24 (2006), p. 243116 (cit. on p. 94).

[Far+97] T. H. Farquhar, A. Chatziioannou, G. Chinn, M. Dahlbom, and E. J. Hoffman. “An investigation of filter choice for filtered back-projection reconstruction in PET”. In: *IEEE Nuclear Science Symposium, 1997*. Vol. 2. IEEE. 1997, pp. 1042–1046 (cit. on pp. 5, 15, 45, 69).

[FDK84] L. A. Feldkamp, L. C. Davis, and J. W. Kress. “Practical cone-beam algorithm”. *Journal of the Optical Society of America A* 1.6 (1984), pp. 612–619 (cit. on pp. 34, 35, 91).

[Fil+09] L. Filion, M. Marechal, B. van Oorschot, D. Pelt, F. Smallenburg, and M. Dijkstra. “Efficient method for predicting crystal structures at finite temperature: Variable box shape simulations”. *Physical review letters* 103.18 (2009), p. 188302 (cit. on p. 132).

[Fra92] J. Frank. *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*. Plenum Press, 1992 (cit. on p. 93).

[GB08] J. Gregor and T. Benson. “Computational Analysis and Improvement of SIRT”. *IEEE Transactions on Medical Imaging* 27.7 (2008), pp. 918–924 (cit. on p. 12).

[GBH70] R. Gordon, R. Bender, and G. T. Herman. “Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography”. *Journal of Theoretical Biology* 29.3 (1970), pp. 471–481 (cit. on p. 7).

[GGL12] A. Guerrero-Martínez, M. Grzelczak, and L. M. Liz-Marzán. “Molecular thinking for nanoplasmonic design”. *ACS nano* 6.5 (2012), pp. 3655–3662 (cit. on p. 94).

[Gib+15] J. W. Gibbs, K. A. Mohan, E. B. Gulsoy, A. J. Shahani, X. Xiao, C. A. Bouman, M. De Graef, and P. W. Voorhees. “The Three-Dimensional Morphology of Growing Dendrites”. *Scientific Reports* 5 (2015), p. 11824 (cit. on p. 107).

[Gil72] P. Gilbert. “Iterative methods for the three-dimensional reconstruction of an object from projections”. *Journal of Theoretical Biology* 36.1 (1972), pp. 105–117 (cit. on p. 95).

[GKP00] M. Grass, T. Köhler, and R. Proksa. “3D cone-beam CT reconstruction for circular trajectories”. *Physics in Medicine and Biology* 45.2 (2000), p. 329 (cit. on p. 44).

[Gor+11] B. Goris, S. Bals, W. Van den Broek, J. Verbeeck, and G. Van Tendeloo. “Exploring different inelastic projection mechanisms for electron tomography”. *Ultramicroscopy* 111.8 (2011), pp. 1262–1267 (cit. on p. 94).

[Gor+12a] B. Goris, S. Bals, W. Van den Broek, E. Carbó-Argibay, S. Gómez-Graña, L. M. Liz-Marzán, and G. Van Tendeloo. “Atomic-scale determination of surface facets in gold nanorods”. *Nature materials* 11.11 (2012), pp. 930–935 (cit. on p. 94).

[Gor+12b] B. Goris, W. Van den Broek, K. J. Batenburg, H. H. Mezerji, and S. Bals. “Electron tomography based on a total variation minimization reconstruction technique”. *Ultramicroscopy* 113 (2012), pp. 120–130 (cit. on p. 95).

[Gra13] P. Grangeat. *Tomography*. Wiley, 2013 (cit. on p. 11).

[Grz+08] M. Grzelczak, J. Pérez-Juste, P. Mulvaney, and L. M. Liz-Marzán. “Shape control in gold nanoparticle synthesis”. *Chemical Society Reviews* 37.9 (2008), pp. 1783–1791 (cit. on p. 93).

[GS07] S. C. Glotzer and M. J. Solomon. “Anisotropy of building blocks and their assembly into complex structures”. *Nature materials* 6.8 (2007), pp. 557–562 (cit. on p. 94).

[Gür+14] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen. “TomoPy: a framework for the analysis of synchrotron tomographic data”. *Journal of Synchrotron Radiation* 21.5 (2014), pp. 1188–1193 (cit. on pp. 10, 107, 108).

[Gür+15a] D. Gürsoy, T. Biçer, J. D. Almer, R. Kettimuthu, S. R. Stock, and F. De Carlo. "Maximum a posteriori estimation of crystallographic phases in X-ray diffraction tomography". *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 373.2043 (2015), p. 20140392 (cit. on p. 109).

[Gür+15b] D. Gürsoy, T. Biçer, A. Lanzirotti, M. G. Newville, and F. De Carlo. "Hyperspectral image reconstruction for x-ray fluorescence tomography". *Optics Express* 23.7 (2015), pp. 9014–9023 (cit. on p. 109).

[Hay09] S. S. Haykin. *Neural Networks: A Comprehensive Foundation 3rd Ed.* Prentice-Hall Of India Pvt. Limited, 2009 (cit. on pp. 71, 76).

[HM94] M. T. Hagan and M. B. Menhaj. "Training feedforward networks with the Marquardt algorithm". *IEEE Transactions on Neural Networks* 5.6 (1994), pp. 989–993 (cit. on p. 77).

[HS52] M. R. Hestenes and E. Stiefel. "Methods of conjugate gradients for solving linear systems". *Journal of Research of the National Bureau of Standards* 49.6 (1952), pp. 409–436 (cit. on p. 7).

[Jan+15] E. Janssens, D. M. Pelt, J. De Beenhouwer, M. van Dael, P. Verboven, B. Nicolai, and J. Sijbers. "Fast Neural Network Based X-ray Tomography of Fruit on a Conveyor Belt". In: *Chemical Engineering Transactions*. Vol. 44. AIDIC. 2015, pp. 181–186 (cit. on p. 131).

[Kac37] S. Kaczmarz. "Angenäherte Auflösung von Systemen linearer Gleichungen". *Bulletin International de l'Academie Polonaise des Sciences et des Lettres* 35 (1937), pp. 355–357 (cit. on p. 7).

[Kat+10] H. Katz-Boon, C. J. Rossouw, M. Weyland, A. M. Funston, P. Mulvaney, and J. Etheridge. "Three-dimensional morphology and crystallography of gold nanorods". *Nano letters* 11.1 (2010), pp. 273–278 (cit. on p. 93).

[KB95a] J. P. Kerr and E. B. Bartlett. "Medical image processing utilizing neural networks trained on a massively parallel computer". *Computers in Biology and Medicine* 25.4 (1995), pp. 393–403 (cit. on p. 67).

[KB95b] J. P. Kerr and E. B. Bartlett. "A statistically tailored neural network approach to tomographic image reconstruction". *Medical physics* 22.5 (1995), pp. 601–610 (cit. on p. 67).

[Kos+00] A. J. Koster, U. Ziese, A. J. Verkleij, A. H. Janssen, and K. P. De Jong. "Three-dimensional transmission electron microscopy: a novel imaging and characterization technique with nanometer scale resolution for materials science". *The Journal of Physical Chemistry B* 104.40 (2000), pp. 9368–9370 (cit. on p. 94).

[Kos+13] A. Kostenko, K. J. Batenburg, A. King, S. E. Offerman, and L. J. van Vliet. "Total variation minimization approach in in-line x-ray phase-contrast tomography". *Optics Express* 21.10 (2013), pp. 12185–12196 (cit. on pp. 8, 42).

[Kos+97] A. J. Koster, R. Grimm, D. Typke, R. Hegerl, A. Stoschek, J. Walz, and W. Baumeister. “Perspectives of molecular and cellular electron tomography”. *Journal of Structural Biology* 120.3 (1997), pp. 276–308 (cit. on p. 93).

[KS01] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001 (cit. on pp. 5, 11, 15, 17, 35, 36, 41, 45, 46, 53, 65, 66, 69, 73, 80, 91).

[KSK00] M. Kachelrieß, S. Schaller, and W. A. Kalender. “Advanced single-slice rebinning in cone-beam spiral CT”. *Medical physics* 27.4 (2000), pp. 754–772 (cit. on p. 44).

[Kun+07] H. Kunze, W. Haerer, J. Orman, T. Mertelmeier, and K. Stierstorfer. “Filter determination for tomosynthesis aided by iterative reconstruction techniques”. In: *9th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. 2007, pp. 309–312 (cit. on p. 13).

[Lan51] L. Landweber. “An Iteration Formula for Fredholm Integral Equations of the First Kind”. *American Journal of Mathematics* 73 (1951), pp. 615–624 (cit. on pp. 6, 17, 36, 46).

[LE00] S. Link and M. A. El-Sayed. “Shape and size dependence of radiative, non-radiative and photothermal properties of gold nanocrystals”. *International Reviews in Physical Chemistry* 19.3 (2000), pp. 409–453 (cit. on p. 105).

[Lov+13] G. Lovric, S. F. Barré, J. C. Schittny, M. Roth-Kleiner, M. Stampanoni, and R. Mokso. “Dose optimization approach to fast X-ray microtomography of the lung alveoli”. *Journal of Applied Crystallography* 46.4 (2013), pp. 856–860 (cit. on pp. 2, 41, 42, 107).

[Lud+08] J. Ludwig, T. Mertelmeier, H. Kunze, and W. Härer. “A novel approach for filtered backprojection in tomosynthesis based on filter kernels determined by iterative reconstruction techniques”. In: *Digital Mammography*. Springer, 2008, pp. 612–620 (cit. on p. 13).

[Mar63] D. W. Marquardt. “An algorithm for least-squares estimation of nonlinear parameters”. *Journal of the Society for Industrial & Applied Mathematics* 11.2 (1963), pp. 431–441 (cit. on p. 77).

[MB12] P. A. Midgley and S. Bals. “Electron Tomography”. In: *Handbook of Nanoscopy*. Ed. by G. van Tendeloo, D. van Dyck, and S. J. Pennycook. Wiley-VCH Verlag GmbH & Co. KGaA, 2012, pp. 253–279 (cit. on p. 94).

[MD09] P. A. Midgley and R. E. Dunin-Borkowski. “Electron tomography and holography in materials science”. *Nature materials* 8.4 (2009), pp. 271–280 (cit. on p. 93).

[MDG95] B. F. McEwen, K. H. Downing, and R. M. Glaeser. “The relevance of dose-fractionation in tomography of radiation-sensitive specimens”. *Ultramicroscopy* 60.3 (1995), pp. 357–373 (cit. on pp. 12, 42, 66).

[Miq+14] E. X. Miqueles, J. Rinkel, F. O'Dowd, and J. S. V. Bermúdez. "Generalized Titarenko's algorithm for ring artefacts reduction". *Journal of Synchrotron Radiation* 21.6 (2014), pp. 1333–1346 (cit. on p. 109).

[Mok+13] R. Mokso, F. Marone, S. Irvine, M. Nyvlt, D. Schwyn, K. Mader, G. K. Taylor, H. G. Krapp, M. Skeren, and M. Stampanoni. "Advantages of phase retrieval for fast x-ray tomographic microscopy". *Journal of Physics D: Applied Physics* 46.49 (2013), p. 494004 (cit. on pp. 12, 42, 107).

[Moo+13] J. Moosmann, A. Ershov, V. Altapova, T. Baumbach, M. S. Prasad, C. LaBonne, X. Xiao, J. Kashef, and R. Hofmann. "X-ray phase-contrast in vivo microtomography probes new aspects of Xenopus gastrulation". English. *Nature* 497.7449 (2013), pp. 374–377 (cit. on p. 107).

[MS12] F. Marone and M. Stampanoni. "Regridding reconstruction algorithm for real-time tomographic imaging". *Journal of Synchrotron Radiation* 19.6 (2012), pp. 1029–1037 (cit. on p. 115).

[Mün+09] B. Münch, P. Trtik, F. Marone, and M. Stampanoni. "Stripe and ring artifact removal with combined wavelet—Fourier filtering". *Optics Express* 17.10 (2009), pp. 8567–8591 (cit. on pp. 109, 116).

[MVB11] H. H. Mezerji, W. Van den Broek, and S. Bals. "A practical method to determine the effective resolution in incoherent experimental electron tomography". *Ultramicroscopy* 111.5 (2011), pp. 330–336 (cit. on p. 95).

[MW03] P. A. Midgley and M. Weyland. "3D electron microscopy in the physical sciences: the development of Z-contrast and EFTEM tomography". *Ultramicroscopy* 96.3 (2003), pp. 413–431 (cit. on pp. 93, 94).

[MWC15] J. C. E. Mertens, J. J. Williams, and N. Chawla. "A method for zinger artifact reduction in high-energy x-ray computed tomography". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 800 (2015), pp. 82–92 (cit. on p. 109).

[MXN07] K. Mueller, F. Xu, and N. Neophytou. "Why do commodity graphics hardware boards (GPUs) work so well for acceleration of computed tomography?" In: *Electronic Imaging 2007*. International Society for Optics and Photonics. 2007, 64980N–64980N (cit. on pp. 4, 45).

[Nat01] F. Natterer. *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, 2001 (cit. on pp. 11, 41, 65).

[Nie+12] T. Nielsen, S. Hitziger, M. Grass, and A. Iske. "Filter calculation for x-ray tomosynthesis reconstruction". *Physics in Medicine and Biology* 57.12 (2012), p. 3915 (cit. on p. 12).

[NM65] J. A. Nelder and R. Mead. "A simplex method for function minimization". *The computer journal* 7.4 (1965), pp. 308–313 (cit. on p. 54).

[NPK10] Z. Nie, A. Petukhova, and E. Kumacheva. "Properties and emerging applications of self-assembled structures made from inorganic nanoparticles". *Nature nanotechnology* 5.1 (2010), pp. 15–25 (cit. on p. 94).

[NW90] D. Nguyen and B. Widrow. “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights”. In: *International Joint Conference on Neural Networks*. Vol. 3. 1990, pp. 21–26 (cit. on pp. 77, 78).

[Oli07] T. E. Oliphant. “Python for scientific computing”. *Computing in Science & Engineering* 9.3 (2007), pp. 10–20 (cit. on pp. 23, 79).

[Pag+02] D. Paganin, S. C. Mayo, T. E. Gureyev, P. R. Miller, and S. W. Wilkins. “Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object”. *Journal of Microscopy* 206.1 (2002), pp. 33–40 (cit. on p. 109).

[Pan+11] W.-M. Pang, J. Qin, Y. Lu, Y. Xie, C.-K. Chui, and P.-A. Heng. “Accelerating simultaneous algebraic reconstruction technique with motion compensation using CUDA-enabled GPU”. *International journal of computer assisted radiology and surgery* 6.2 (2011), pp. 187–199 (cit. on p. 12).

[Pat+16] B. M. Patterson, N. L. Cordes, K. Henderson, J. J. Williams, T. Stannard, S. S. Singh, A. R. Ovejero, X. Xiao, M. Robinson, and N. Chawla. “In situ X-ray synchrotron tomographic imaging during the compression of hyper-elastic polymeric materials”. English. *Journal of Materials Science* 51.1 (2016), pp. 171–187 (cit. on p. 107).

[PB13] D. M. Pelt and K. J. Batenburg. “Fast tomographic reconstruction from limited data using artificial neural networks”. *IEEE Transactions on Image Processing* 22.12 (2013), pp. 5238–5251 (cit. on pp. 65, 131).

[PB14a] D. M. Pelt and K. J. Batenburg. “Improving Filtered Backprojection Reconstruction by Data-Dependent Filtering”. *IEEE Transactions on Image Processing* 23.11 (2014), pp. 4750–4762 (cit. on pp. 11, 131).

[PB14b] D. M. Pelt and R. H. Bisseling. “A medium-grain method for fast 2D bipartitioning of sparse matrices”. In: *IEEE 28th International Parallel and Distributed Processing Symposium 2014*. IEEE. 2014, pp. 529–539 (cit. on p. 132).

[PB15a] D. M. Pelt and K. J. Batenburg. “Accurately approximating algebraic tomographic reconstruction by filtered backprojection”. In: *Proceedings of the 2015 International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Ed. by M. King, S. Glick, and K. Mueller. 2015, pp. 158–161 (cit. on pp. 35, 131).

[PB15b] D. M. Pelt and R. H. Bisseling. “An exact algorithm for sparse matrix bipartitioning”. *Journal of Parallel and Distributed Computing* 85 (2015), pp. 79–90 (cit. on p. 131).

[PBon] D. M. Pelt and K. J. Batenburg. “A method for locally approximating advanced regularized iterative tomographic reconstruction methods”. *IEEE Transactions on Image Processing* (Submitted for publication) (cit. on pp. 41, 131).

[PBS11] W. J. Palenstijn, K. J. Batenburg, and J. Sijbers. “Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs)”. *Journal of Structural Biology* 176.2 (2011), pp. 250–253 (cit. on pp. 4, 19, 23, 38, 44, 54, 80, 109).

[Pec+08] C. Pecharromán, J. Pérez-Juste, G. Mata-Osoro, L. M. Liz-Marzán, and P. Mulvaney. “Redshift of surface plasmon modes of small gold rods due to their atomic roughness and end-cap geometry”. *Physical Review B* 77.3 (2008), p. 035418 (cit. on p. 93).

[Pel+on] D. M. Pelt, D. Gürsoy, W. J. Palenstijn, J. Sijbers, F. De Carlo, and K. J. Batenburg. “Integration of TomoPy and the ASTRA toolbox for advanced processing and reconstruction of tomographic synchrotron data”. *Journal of Synchrotron Radiation* (Submitted for publication) (cit. on pp. 107, 131).

[Pér+05] J. Pérez-Juste, I. Pastoriza-Santos, L. M. Liz-Marzán, and P. Mulvaney. “Gold nanorods: synthesis, characterization and applications”. *Coordination Chemistry Reviews* 249.17 (2005), pp. 1870–1901 (cit. on p. 93).

[Per+10] E. M. Perassi, J. C. Hernandez-Garrido, M. S. Moreno, E. R. Encina, E. A. Coronado, and P. A. Midgley. “Using highly accurate 3D nanometrology to model the optical properties of highly irregular nanoparticles: a powerful tool for rational design of plasmonic devices”. *Nano letters* 10.6 (2010), pp. 2097–2104 (cit. on p. 105).

[PG15] C. Phatak and D. Gürsoy. “Iterative reconstruction of magnetic induction using Lorentz transmission electron tomography”. *Ultramicroscopy* 150 (2015), pp. 54–64 (cit. on p. 109).

[PL13] S. Peetermans and E. H. Lehmann. “Simultaneous neutron transmission and diffraction contrast tomography as a non-destructive 3D method for bulk single crystal quality investigations”. *Journal of Applied Physics* 114.12 (2013), p. 124905 (cit. on p. 109).

[Pla+15] L. Plantagie, W. van Aarle, J. Sijbers, and K. J. Batenburg. “Filtered backprojection using algebraic filters; application to biomedical micro-CT data”. In: *IEEE 12th International Symposium on Biomedical Imaging (ISBI), 2015*. IEEE. 2015, pp. 1596–1599 (cit. on p. 109).

[PSB13] D. M. Pelt, J. Sijbers, and K. J. Batenburg. “Fast Tomographic Reconstruction From Highly Limited Data Using Artificial Neural Networks”. In: *Proceedings of 1st International Conference on Tomography of Materials and Structures 2013 (ICTMS 1)*. Ed. by V. Cnudde and D. Bernard. University Press, 2013, pp. 109–112 (cit. on p. 131).

[PSV09] X. Pan, E. Y. Sidky, and M. Vannier. “Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?” *Inverse problems* 25.12 (2009), p. 123009 (cit. on pp. 5, 11, 12, 36, 42).

[Rei+13] P. Reischig, A. King, L. Nervo, N. Vigano, Y. Guilhem, W. J. Palenstijn, K. J. Batenburg, M. Preuss, and W. Ludwig. “Advances in X-ray diffraction contrast tomography: flexibility in the setup geometry and application to multiphase materials”. *Journal of Applied Crystallography* 46.2 (2013), pp. 297–311 (cit. on p. 109).

[RM05] N. L. Rosi and C. A. Mirkin. “Nanostructures in biodiagnostics”. *Chemical Reviews* 105.4 (2005), pp. 1547–1562 (cit. on p. 93).

[Rod+01] A. F. Rodriguez, W. E. Blass, J. H. Missimer, and K. L. Leenders. “Artificial neural network Radon inversion for image reconstruction”. *Medical physics* 28.4 (2001), pp. 508–514 (cit. on p. 67).

[Roe+12] T. Roelandts, K. J. Batenburg, E. Biermans, C. Kübel, S. Bals, and J. Sijbers. “Accurate segmentation of dense nanoparticles by partially discrete electron tomography”. *Ultramicroscopy* 114 (2012), pp. 96–105 (cit. on p. 109).

[SA10] A. Sakdinawat and D. Attwood. “Nanoscale X-ray imaging”. *Nature photonics* 4.12 (2010), pp. 840–848 (cit. on p. 107).

[Sal+03] L. Salvo, P. Cloetens, E. Maire, S. Zabler, J. Blandin, J.-Y. Buffière, W. Ludwig, E. Boller, D. Bellet, and C. Josserond. “X-ray micro-tomography an attractive characterisation technique in materials science”. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 200 (2003), pp. 273–286 (cit. on p. 2).

[Sán+06] A. Sánchez-Iglesias, I. Pastoriza-Santos, J. Pérez-Juste, B. Rodríguez-González, F. J. García de Abajo, and L. M. Liz-Marzán. “Synthesis and optical properties of gold nanodecahedra with size control”. *Advanced Materials* 18.19 (2006), pp. 2529–2534 (cit. on p. 93).

[Sco+12] M. C. Scott, C.-C. Chen, M. Mecklenburg, C. Zhu, R. Xu, P. Ercius, U. Dahmen, B. C. Regan, and J. Miao. “Electron tomography at 2.4-ångström resolution”. *Nature* 483.7390 (2012), pp. 444–447 (cit. on p. 2).

[SH10] Z. Shi and L. He. “Application of neural networks in medical image processing”. In: *Proceedings of the Second International Symposium on Networking and Network Security*. 2010, pp. 2–4 (cit. on p. 66).

[SHO93] V. Srinivasan, Y. K. Han, and S. H. Ong. “Image reconstruction by a Hopfield neural network”. *Image and Vision Computing* 11.5 (1993), pp. 278–282 (cit. on p. 66).

[Sip93] H. Sipila. “Moving-object computer tomography for luggage inspection”. In: *Applications in Optical Science and Engineering*. International Society for Optics and Photonics. 1993, pp. 39–40 (cit. on pp. 2, 12, 66).

[SP08] E. Y. Sidky and X. Pan. “Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization”. *Physics in Medicine and Biology* 53.17 (2008), p. 4777 (cit. on pp. 8, 17, 21, 42, 47, 66).

[THY08] A. R. Tao, S. Habas, and P. Yang. “Shape Control of Colloidal Metal Nanocrystals”. *Small* 4.3 (2008), pp. 310–325 (cit. on p. 93).

[TLL95] I. V. Tetko, D. J. Livingstone, and A. I. Luik. “Neural network studies. 1. Comparison of overfitting and overtraining”. *Journal of Chemical Information and Computer Sciences* 35.5 (1995), pp. 826–833 (cit. on p. 71).

[Van+11] S. Van Aert, K. J. Batenburg, M. D. Rossell, R. Erni, and G. Van Tendeloo. “Three-dimensional atomic imaging of crystalline nanoparticles”. *Nature* 470.7334 (2011), pp. 374–377 (cit. on p. 94).

[Van+15] G. Van Eyndhoven, K. J. Batenburg, D. Kazantsev, V. Van Nieuwenhove, P. D. Lee, K. J. Dobson, and J. Sijbers. “An Iterative CT Reconstruction Algorithm for Fast Fluid Flow Imaging”. *IEEE Transactions on Image Processing* 24.11 (2015), pp. 4446–4458 (cit. on p. 109).

[VKB11] J. K. Vis, W. A. Kosters, and K. J. Batenburg. “Discrete tomography: A neural network approach”. In: *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC)*. 2011, pp. 328–335 (cit. on p. 67).

[Vo+14] N. T. Vo, M. Drakopoulos, R. C. Atwood, and C. Reinhard. “Reliable method for calculating the center of rotation in parallel-beam tomography”. *Optics Express* 22.16 (2014), pp. 19078–19086 (cit. on p. 109).

[Wan+04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612 (cit. on pp. 24, 39, 54).

[Wey+06] M. Weyland, T. J. V. Yates, R. E. Dunin-Borkowski, L. Laffont, and P. A. Midgley. “Nanoscale analysis of three-dimensional structures by electron tomography”. *Scripta Materialia* 55.1 (2006), pp. 29–33 (cit. on p. 94).

[Wol+10] D. Wolf, A. Lubk, H. Lichte, and H. Friedrich. “Towards automated electron holographic tomography for 3D mapping of electrostatic potentials”. *Ultramicroscopy* 110.5 (2010), pp. 390–399 (cit. on p. 94).

[WP05] R. C. Whaley and A. Petitet. “Minimizing development and maintenance costs in supporting persistently optimized BLAS”. *Software: Practice and Experience* 35.2 (2005), pp. 101–121 (cit. on pp. 23, 79).

[WW97] Y. Wang and F. M. Wahl. “Vector-entropy optimization-based neural-network approach to image reconstruction from projections”. *IEEE Transactions on Neural Networks* 8.5 (1997), pp. 1008–1014 (cit. on p. 66).

[WWH05] Y. Wei, G. Wang, and J. Hsieh. “An intuitive discussion on the ideal ramp filter in computed tomography (I)”. *Computers & Mathematics with Applications* 49.5 (2005), pp. 731–740 (cit. on pp. 5, 15).

[XM05] F. Xu and K. Mueller. “Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware”. *IEEE Transactions on Nuclear Science* 52.3 (2005), pp. 654–663 (cit. on pp. 4, 12, 17, 42, 45, 66).

[XM06] F. Xu and K. Mueller. “A comparative study of popular interpolation and integration methods for use in computed tomography”. In: *3^{rd} IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006*. IEEE. 2006, pp. 1252–1255 (cit. on pp. 53, 110).

[Yeg09] B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009 (cit. on p. 71).

[ZCG09] P. Zijlstra, J. W. Chon, and M. Gu. “Five-dimensional optical recording mediated by surface plasmons in gold nanorods”. *Nature* 459.7245 (2009), pp. 410–413 (cit. on p. 93).

[Zen12] G. L. Zeng. “A filtered backprojection algorithm with characteristics of the iterative Landweber algorithm”. *Medical physics* 39.2 (2012), pp. 603–607 (cit. on pp. 12, 36, 43).

[ZO11] P. Zijlstra and M. Orrit. “Single metal nanoparticles: optical detection, spectroscopy and applications”. *Reports on Progress in Physics* 74.10 (2011), p. 106401 (cit. on p. 93).

[ZZ13] G. L. Zeng and A. Zamyatin. “A filtered backprojection algorithm with ray-by-ray noise weighting”. *Medical physics* 40.3 (2013), p. 031113 (cit. on pp. 12, 43).

List of publications

- Publications that are part of this thesis:
    - D. M. Pelt, D. Gürsoy, W. J. Palenstijn, J. Sijbers, F. De Carlo, and K. J. Batenburg. "Integration of TomoPy and the ASTRA toolbox for advanced processing and reconstruction of tomographic synchrotron data". *Journal of Synchrotron Radiation* (Submitted for publication).
    - D. M. Pelt and K. J. Batenburg. "A method for locally approximating advanced regularized iterative tomographic reconstruction methods". *IEEE Transactions on Image Processing* (Submitted for publication).
    - E. Bladt, D. M. Pelt, S. Bals, and K. J. Batenburg. "Electron tomography based on highly limited data using a neural network reconstruction technique". *Ultramicroscopy* 158 (2015), pp. 81–88.
    - D. M. Pelt and K. J. Batenburg. "Accurately approximating algebraic tomographic reconstruction by filtered backprojection". In: *Proceedings of the 2015 International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Ed. by M. King, S. Glick, and K. Mueller. 2015, pp. 158–161.
    - D. M. Pelt and K. J. Batenburg. "Improving Filtered Backprojection Reconstruction by Data-Dependent Filtering". *IEEE Transactions on Image Processing* 23.11 (2014), pp. 4750–4762.
    - D. M. Pelt and K. J. Batenburg. "Fast tomographic reconstruction from limited data using artificial neural networks". *IEEE Transactions on Image Processing* 22.12 (2013), pp. 5238–5251.
- Publications on the subject of tomography that are not part of this thesis:
    - E. Janssens, D. M. Pelt, J. De Beenhouwer, M. van Dael, P. Verboven, B. Nicolai, and J. Sijbers. "Fast Neural Network Based X-ray Tomography of Fruit on a Conveyor Belt". In: *Chemical Engineering Transactions*. Vol. 44. AIDIC. 2015, pp. 181–186.
    - D. M. Pelt, J. Sijbers, and K. J. Batenburg. "Fast Tomographic Reconstruction From Highly Limited Data Using Artificial Neural Networks". In: *Proceedings of 1st International Conference on Tomography of Materials and Structures 2013 (ICTMS 1)*. Ed. by V. Cnudde and D. Bernard. University Press, 2013, pp. 109–112.
- Publications on other topics that are not part of this thesis:
    - D. M. Pelt and R. H. Bisseling. "An exact algorithm for sparse matrix bipartitioning". *Journal of Parallel and Distributed Computing* 85 (2015), pp. 79–90.

- D. M. Pelt and R. H. Bisseling. “A medium-grain method for fast 2D bipartitioning of sparse matrices”. In: *IEEE 28th International Parallel and Distributed Processing Symposium 2014*. IEEE. 2014, pp. 529–539.
- L. Filion, M. Marechal, B. van Oorschot, D. Pelt, F. Smallenburg, and M. Dijkstra. “Efficient method for predicting crystal structures at finite temperature: Variable box shape simulations”. *Physical review letters* 103.18 (2009), p. 188302.

Samenvatting

In veel gevallen is het nuttig om een manier te hebben om binnenin een object te kunnen kijken zonder te hoeven beschadigen. In de medische wereld, bijvoorbeeld, is het handig om een diagnose te kunnen stellen voor een patiënt zonder te hoeven opereren. In ziekenhuizen wordt hiervoor vaak een röntgenfoto gebruikt. Bij een röntgenfoto wordt de patiënt belicht met een röntgenbron en vervolgens wordt de straling achter de patiënt opgevangen door een detector. Aangezien verschillende delen van het lichaam verschillende hoeveelheden straling absorberen, zal op de detector een beeld van de interne structuur van het lichaam te zien zijn. Een voorbeeld van een röntgenfoto is te zien in Fig. S1.

In een röntgenfoto is het vaak lastig om te zien of iets zich voorin het lichaam (dichter bij de röntgenbron) of achterin het lichaam (dichter bij de detector) bevindt. Meestal kan een specialist toch een diagnose maken op basis van een röntgenfoto, aangezien er veel bekend is over de structuur van het menselijk lichaam. In veel gevallen is het echter noodzakelijk om de complete drie-dimensionale informatie over de interne structuur van de patiënt te hebben. Bij het behandelen van een tumor, bijvoorbeeld, is het belangrijk om de precieze positie van de tumor in het lichaam te weten. Met behulp van *computertomografie* (CT) kan in zulke gevallen een drie-dimensionaal beeld worden gemaakt van de interne structuur van een patiënt.

In computertomografie worden de röntgenbron en de detector om de patiënt heen gedraaid. Op deze manier worden er meerdere röntgenfoto's gemaakt, elk onder een andere hoek. Een schematisch overzicht van een computertomografie scanner is afgebeeld in Fig. S2a. De opgenomen twee-dimensionale röntgenfoto's, ook wel *projecties* genoemd in deze context, worden gebruikt om een drie-dimensionaal beeld van de interne structuur te berekenen door middel van een wiskundig *reconstructie algoritme*. Een voorbeeld van een plakje van een gereconstrueerd beeld van een patiënt is afgebeeld in Fig. S2b.

Tomografie wordt in veel meer toepassingen gebruikt dan alleen in de medische wereld. In materiaalwetenschappen is het bijvoorbeeld nuttig om binnenin verschillende materialen te kunnen kijken terwijl ze onder druk staan of verhit worden. Een ander voorbeeld is de paleontologie, waar het nuttig is om de drie-dimensionale structuur van

Figuur S1: Röntgenfoto van een menselijke schouder (© Nevit Dilmen).

(a) (b)

Figuur S2: (a) Schematisch overzicht van een computertomografie (CT) scanner. (b) Een plakje van een gereconstrueerde tomografische dataset (bron: James Heilman, MD).

fossielen te kunnen bestuderen zonder ze te hoeven beschadigen. Andere voorbeelden van toepassingen van tomografie zijn biomedische onderzoeken en het controleren en optimaliseren van productieprocessen in industiële toepassingen. Verschillende stralingsbronnen en scanners kunnen worden gebruikt om tomografie mee uit te voeren. In dit proefschrift wordt vooral aandacht besteed aan twee veelgebruikte stralingsbronnen: synchrotrons en elektronenmicroscopen. In synchrotrons wordt extreem felle röntgenstraling opgewekt door deeltjes te versnellen in een ring die enkele kilometers lang kan zijn. In een elektronenmicroscoop wordt een bundel van elektronen gebruikt om een beeld te vormen in plaats van een bundel van röntgenstralen. Het voordeel hiervan is dat je erg kleine deeltjes kunt scannen, tot op de nanometer schaal.

Een belangrijk onderdeel van elke toepassing van tomografie is het wiskundige reconstructie algoritme. Het doel van een reconstructie algoritme is het berekenen van een drie-dimensionaal beeld van het gescande object uit de opgenomen tweedimensionale projecties. Er bestaan veel verschillende algoritmes voor dit probleem, welke ruwweg opgedeeld kunnen worden in twee groepen: *analytische* en *algebraïsche* algoritmes. Analytische algoritmes hebben meestal een filterstap, waarin de opgenomen projecties worden getransformeerd met behulp van een *filter*. Een voordeel van analytische algoritmes is dat ze snel uit te rekenen zijn, vaak sneller dan dat de projecties opgenomen kunnen worden. Een belangrijk nadeel is dat beelden berekend met analytische algoritmes vaak veel fouten bevatten als er weinig projecties beschikbaar zijn of als er ruis aanwezig is in de projecties, wat vaak het geval is in moderne geavanceerde tomografie experimenten. Algebraïsche algoritmes kunnen in deze gevallen vaak betere beelden berekenen door gebruik te maken van voorkennis over het gemeten object of de gebruikte scanner. Een groot nadeel van algebraïsche algoritmes is echter dat ze veel langzamer zijn dan analytische algoritmes, vaak zo langzaam dat ze in de praktijk in veel gevallen onbruikbaar blijken te zijn.

In dit proefschrift worden nieuwe reconstructie algoritmes geïntroduceerd genaamd *filter-gebaseerde algoritmes*. Deze algoritmes gebruiken een combinatie van de analytische en de algebraïsche aanpak. De snelheid van de nieuwe algoritmes is vergelijkbaar met die van analytische algoritmes, maar de beeldkwaliteit is vergelijkbaar met die van algebraïsche algoritmes. Bij de nieuwe algoritmes wordt gebruik gemaakt van het feit dat er in de filterstap van analytische algoritmes de vrijheid is om andere filters te gebruiken dan de standaard filters. Deze filters kunnen op verschillende manieren gedefinieerd en berekend worden, en kunnen afhangen van, bijvoorbeeld, de manier waarop gescand is of de eigenschappen van de objecten die gescand zijn. Verschillende filter-gebaseerde algoritmes worden geïntroduceerd in dit proefschrift, en resultaten van de nieuwe algoritmes worden vergeleken met populaire bestaande algoritmes.

In Hoofdstuk 2 wordt het MR-FBP algoritme geïntroduceerd. Dit algoritme gebruikt een filter dat berekend wordt op een manier die lijkt op algebraïsche reconstructie algoritmes. Het resultaat is dat het berekende filter afhangt van de data die opgenomen is, en daardoor geoptimaliseerd is voor de scanner die gebruikt is en het object dat gescand is. Resultaten uit Hoofdstuk 2 laten zien dat de beeldkwaliteit van MR-FBP vergelijkbaar is met die van een populair algebraïsch algoritme, maar dat MR-FBP significant sneller beelden kan berekenen. Ook laten de resultaten zien dat het berekende filter zich automatisch aanpast aan de gescande objecten en de gebruikte scanner.

Het SIRT-FBP algoritme wordt geïntroduceerd in Hoofdstuk 3. Hierbij wordt een specifiek algebraïsch algoritme, genaamd SIRT, benaderd door een analytisch algoritme met een speciaal filter. Door de wiskundige beschrijving van het SIRT algoritme te herschrijven wordt in Hoofdstuk 3 aangetoond dat SIRT gelijkenissen vertoont met analytische algoritmes. Vervolgens worden de herschreven formules gebruikt om speciale filters te berekenen waarmee SIRT benaderd kan worden door middel van een analytisch algoritme. De speciale filters kunnen voorberekend worden voor de gekozen instellingen van de scanner, waarna elk reconstructie beeld snel berekend kan worden. Resultaten uit Hoofdstuk 3 laten zien dat de beelden berekend door het SIRT-FBP algoritme vrijwel identiek zijn aan de beelden berekend door het SIRT algoritme.

In Hoofdstuk 4 wordt het SIRT-FBP algoritme uitgebreid waardoor het mogelijk wordt om voorkennis over het gescande object te gebruiken om de beeldkwaliteit te verbeteren. Bestaande algebraïsche algoritmes die voorkennis kunnen gebruiken zijn vaak erg langzaam, en moeten altijd een beeld berekenen van het hele gescande object, zelfs als alleen een klein gebied interessant is. Het resultaat van Hoofdstuk 4 is een algoritme waarmee snel een beeld van hoge kwaliteit kan worden berekend voor een klein gebied van het gescande object. Resultaten laten zien dat de beeldkwaliteit van het nieuwe algoritme vrijwel identiek is aan die van bestaande algebraïsche algoritmes, maar dat de beelden veel sneller berekend kunnen worden door het nieuwe algoritme.

In Hoofdstuk 5 wordt kunstmatige intelligentie gebruikt om filters te bepalen, wat leidt tot het NN-FBP algoritme. Door beelden van hoge kwaliteit te laten zien aan het NN-FBP algoritme tijdens een trainingsfase, leert het algoritme welke filters tot een goede beeldkwaliteit leiden voor specifieke objecten en scanners. Na de trainingsfase kunnen beelden van hoge kwaliteit berekend worden voor nieuw gescande objecten door verschillende filters te combineren. Het NN-FBP algoritme is gebaseerd op een analytisch algoritme, waardoor de rekentijd kort is. Resultaten uit Hoofdstuk 5 laten zien dat met deze aanpak een significant hogere beeldkwaliteit kan worden verkregen vergeleken met bestaande algoritmes. In Hoofdstuk 6 wordt het NN-FBP algoritme toegepast op elektronenmicroscopie, met als doel om het aantal projecties dat nodig is om een goed beeld te krijgen te verminderen ten opzichte van bestaande algoritmes. Hierdoor kunnen er in dezelfde tijd meer objecten gescand worden, waardoor het makkelijker is om statistisch significante informatie te verkrijgen over de gescande objecten.

In Hoofdstuk 7 wordt een ander probleem dan in de andere hoofdstukken behandeld. Een bijkomende reden dat algebraïsche algoritmes niet vaak gebruikt worden is praktisch van aard: het is vaak moeilijk om software dat algebraïsche algoritmes bevat te installeren en te gebruiken bij verschillende experimentele faciliteiten. In Hoofdstuk 7 proberen we deze situatie te verbeteren voor synchrotrons door twee bestaande software pakketten te combineren: TomoPy, een pakket speciaal voor het verwerken van tomografie datasets opgenomen door synchrotrons, en de ASTRA toolbox, een pakket speciaal voor het ontwikkelen van geavanceerde reconstructie algoritmes. Het resultaat van het combineren van de pakketten is dat nieuwe (algebraïsche) algoritmes die ontwikkeld zijn met de ASTRA toolbox gemakkelijk kunnen worden geïnstalleerd en gebruikt bij synchrotrons. In Hoofdstuk 7 laten we zien dat een gebruiker weinig hoeft aan te passen om gebruik te maken van het gecombineerde pakket, en dat door gebruik te maken van de ASTRA toolbox minder rekentijd nodig is voor algebraïsche algoritmes vergeleken met de bestaande algoritmes in TomoPy.

Curriculum Vitae

Daniël M. Pelt was born in Utrecht in 1986 and completed his secondary education (gymnasium) in 2004 at the Christelijk Gymnasium Utrecht. In 2008, he obtained his bachelor's degree in Physics and Astronomy from Utrecht University. He obtained his master's degree (cum laude) in Mathematics from Utrecht University in 2010, writing a thesis entitled "Matrix Partitioning: Optimal bipartitioning and heuristic solutions" under supervision of prof. dr. R. H. Bisseling. After graduating, he worked as a Scientific Programmer at the department of Biological Psychology of VU University in Amsterdam, developing software for the analysis of behavioral and physiological data. In 2012, he started as a PhD student at Centrum Wiskunde & Informatica in Amsterdam, under supervision of prof. dr. K. J. Batenburg. As part of his PhD research, he made extended visits to experimental facilities in France, Switzerland, and Illinois (USA), and presented his work at international workshops and conferences in Ghent, Hong Kong, Phoenix (AZ), Milan, Newport (RI), and London, among others. He is currently employed as a researcher at Centrum Wiskunde & Informatica.

Acknowledgments

This thesis is the result of four years of work, during which I enjoyed the support and help of many people. Here, I would like to express my gratitude to some people in particular.

First and foremost, I would like to thank my supervisor and promotor Joost Batenburg. Your comments and suggestions on how to improve my scientific work have been extremely useful, and I enjoyed the many discussions that we had. Furthermore, I want to thank you for allowing me to work in a way that was the most fitting for me, and for providing the contacts and opportunities to visit other groups and attend conferences and workshops in the Netherlands and abroad.

I would like to thank my colleagues at CWI for creating a stimulating and friendly work atmosphere, which is an important prerequisite for writing a thesis. In particular, I want to thank the people of the tomography group: Jeroen Bédorf, Folkert Bleichrodt, Wagner Fortes, Tristan van Leeuwen, Willem Jan Palenstijn, Linda Plantagie, Paul de Zeeuw, Zhichao Zhong, and Xiaodong Zhuge. The many tomomeetings, fun side projects, and trips together were definitely one of the most enjoyable parts of my PhD project. Special thanks go out to my office mates Folkert Bleichrodt and Jeroen Bédorf for the interesting discussions and for putting up with the growing collection of souvenirs polluting my desk.

This thesis could not have been written without the help of the supporting staff of CWI, especially Nada Mitrovic and Duda Tepsic, for which I am very thankful. I owe thanks to my master thesis supervisor Rob Bisseling for putting me in contact with my PhD supervisor, and for our enjoyable Friday afternoon talks about many different topics. I am also grateful for having had the opportunity to meet many kind people in the international tomography community, with whom I hope to stay in contact in the future. In particular, I would like to thank the people who collaborated with me to produce the work presented in this thesis: Sara Bals, Eva Bladt, Francesco De Carlo, Doğa Gürsoy, and Jan Sijbers.

During my PhD project, I have had the pleasure of enjoying the company of my friends and family. I want to thank my friends and brothers for the many fun evenings, nights, and weekends, which gave me much-needed distractions from my PhD work. I would also like to thank my parents for giving me my curiosity and allowing it to develop during my childhood. I am forever grateful to my mother for her incredible strength and perseverance when faced with the daunting task of raising three boys after my father had passed away.

Finally, I want to give my most special thanks to Hilde for making the process of writing this thesis infinitely more enjoyable. Thank you for sharing your love of traveling with me and reminding me that there is more to life than PhD work. I am glad that we could be part of each other's PhD adventures, and I look forward to many more adventures together in the future.